(2)

Naval Research Laboratory
Washington, DC 20375-5000

NRL Memorandum Report 6014

AD-A183 327

Permittivity Measurements Utilizing the Dielectric Post in Waveguide Technique

FREDERIC J. RACHFORD

Metal Physics Branch
Condensed Matter and Radiation Sciences Division

July 9, 1987

This research was supported by ONR under Contract N00014-85-WR24139 and DARPA under ARPA order 5335

A193 32?

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 6014	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Research Laboratory — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Engineering Logistics Office — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: NELO — PROJECT NO.: 62013N — TASK NO.: R1947 — WORK UNIT ACCESSION NO.:

11. TITLE (Include Security Classification)

Permittivity Measurements Utilizing the Dielectric Post in Waveguide

12. PERSONAL AUTHOR(S)
Rachford, F.J.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM ____ TO ____	1987 July 9	82

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Measurements; Microwave; Dielectric constant measurements; Millimeterwave

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes experimental procedures and computer codes for the measurement of the permittivity of filamentary high loss materials by the dielectric post-in-waveguide (DPW) technique. This method is especially useful in characterizing the microwave/millimeter wave dielectric properties of materials such as partially graphitized carbon fibers, linear chain crystals, superionic conductors, ferroelectrics, etc. This method allows th. complex permittivity, to be measured on a pointwise continuous frequency basis across a rectangular waveguide band. We have employed and will describe two measurement setups; for X and Ku band studies we have collected data with commercial automated vector network analyzers, for K and Ka bands we have assembled transmission phase bridges using standard waveguide components and a scalar network analyzer. The computer codes developed for this work are described in detail and selected programs and subroutines are listed in the appendix.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Frederic J. Rachford	22b. TELEPHONE (Include Area Code) 202-767-4207 — 22c. OFFICE SYMBOL Code 4636

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

PERMITTIVITY MEASUREMENTS UTILIZING THE
DIELECTRIC POST IN WAVEGUIDE TECHNIQUE

SUMMARY

This report describes experimental procedures and computer codes developed at NRL for the measurement of the permittivity of filamentary high loss materials by the dielectric post-in-waveguide (DPW) technique. This method is especially useful in characterizing the microwave/millimeter wave dielectric properties of materials such as partially graphitized carbon fibers, linear chain crystals, superionic conductors, ferroelectrics, etc. This method allows the complex permittivity, to be measured on a pointwise continuous frequency basis across a rectangular waveguide band. We have employed and will describe two measurement setups; for X and Ku band studies we have collected data with commercial automated vector network analyzers, for K and Ka bands we have assembled transmission phase bridges using standard waveguide components and a scalar network analyzer. The data from both measurement configurations are collected on floppy disks for later analysis on a VAX 11/780 computer. The computer codes developed for this work will be described in detail and selected programs and subroutines will be listed in the appendix.

INTRODUCTION

There are many methods of measuring the dielectric constant of materials at microwave frequencies. The simpliest and the most broadband technique employs an automated vector network analyser (AVNA) and a coaxial or waveguide transmission line. The sample is chosen to be somewhat less than a quarter wavelength thick at the lowest measurement frequency and to fill the cross section of the transmission line. The AVNA measures the S-parameters of the loaded transmission line and the frequency dependent permittivity, $\varepsilon(f)$, and permeability, $\mu(f)$, of the sample can be directly calculated from the S-parameters by the AVNA computer/controller. This technique,

Manuscript approved April 14, 1987

however, assumes that the sample is homogeneous in composition and isotropic in microwave dielectric properties. These assumptions fail for highly anisotropic samples. At best, the filled transmission line technique can be employed for measuring anisotropic materials in the special case where the anisotropy axes of the pure material can be aligned with the internal fields of the transmission line or to measure the composite properties of anisotropic materials mechanically dispersed in a binder. In the case of the composite system it is very difficult to infer the intrinsic properties of the anisotropic material from the composite dielectric response.

A second technique, cavity perturbation, is a classic method for measuring the permittivity and permeability of materials and is suited for the study of filamentary samples. The perturbation technique fails for high loss samples which destroy the cavity resonance. Reduction in sample size can alleviate the problem in some cases, however, impossibly small and very fragile samples with large geometric aspect ratios may be required to satisfy the perturbation criterion while minimizing depolarization effects. Perturbation cavities also have the disadvantage of operating at a single frequency.

The dielectric post-in-waveguide method for measuring the permittivity of filamentary or cylindrical post samples shares some of the advantages of the two conventional methods previously mentioned. Like the filled transmission line method it allows relatively broadband automated measurments and like the perturbation cavity is it is adapted for the determination of permittivities of filamentary samples. The technique works best with samples with large dielectric constants and/or high loss - samples which fail in the resonant cavity perturbation technique. Samples chosen for study by this technique must present a sufficient impedance mismatch to the waveguide to measurably affect the reflection and transmission coefficients. Since in this method the cylindrical filament axis of the sample is alligned with the microwave E-field, the permittivity measured is that of the long axis of the sample.

The method was first described by J. Le Bot who applied it to the measurement of ϵ as a function of water absorption in silica and alumina gels (1). Since that time the method has been used relatively infrequently (2,3). In this report we describe how this technique can become a useful tool for research using modern instrumentation and computational capabilities.

The DPW experimental procedure is greatly simiplified by the use of the automated vector network analyser. AVNA's are now commercially available for pointwise continuous swept frequency measurements up to frequencies of 100 GHz. The AVNA available to our group, however, is limited to frequencies less than or equal to 18 GHz. We have thus employed the AVNA for post measurements at X and

Ku bands. Measurements were extended to K band (18 0 26.5 GHz) Ka band (26.5 to 40 GHz) by the construction of simple transmission bridges. In either case the measurement fixture consist of a simple short length of waveguide with a hole drilled thru the center of the broad face. (See Fig. 1). The waveguide sample holder can be heated in a tube furnace or cooled to about 100K by standard gas flow techniques. The swept frequency data is collected on floppy disks on both the AVNA and transmission bridge set-ups and analyzed on a VAX 11/780 computer.

The computer codes created for this analysis employ subroutines which calculate the complex transmission and reflection coefficients for the post geometry. One code calculates the return/insertion loss and phase expected for a given frequency dependent permittivity, sample diameter, and waveguide width. Another program uses the measured return loss and phase, or insertion loss and phase, or scalar return loss and insertion loss to calculate permittivity of a given sample at each measurement frequency across the waveguide band using a Newton inversion technique. A third code performs a least square fit to S-parameter data at each frequency. A fourth code performs at least squares fit to multiple sets of S-parameter data assuming a menu determined frequency dependent functional representation of the complex permittivity of the sample over one or more waveguide bands.

In the following sections we discuss the theory, the experimental set-ups, the computer codes, and provide some examples of the measurement technique. In the appendices we provide a detailed list of the measurement and analysis procedures and list some of the computer codes developed for DPW data reduction.

THEORY

The problem of a dielectric post in a rectangular waveguide was solved by workers of the MIT Radiation Lab during World War II. The method of solution is discussed in detail in a volume of lecture notes from that era by Prof. Julian Schwinger (4) and the results of the calculation to high order were presented in the MIT Waveguide Handbook (5). In most experimental studies employing the post method it is assumed that the sample diameter is very small with respect to the microwave wavelength and waveguide dimensions. In this limit the equations can be simplified somewhat and permittivity may be found using graphical methods (1). Here we assume only that the sample is cylindrical as required by the theory and that it is positioned in the center of the broad face of the waveguide with its cylindrical axis parallel to the microwave E-field.

The dielectric post problem is analyzed in terms of a T shaped equivalent circuit where the two in line impedance elements, Zb, on top of the T are identical and differ from the shunting impedance, Za. The circuit is assumed to be terminated on each end by the

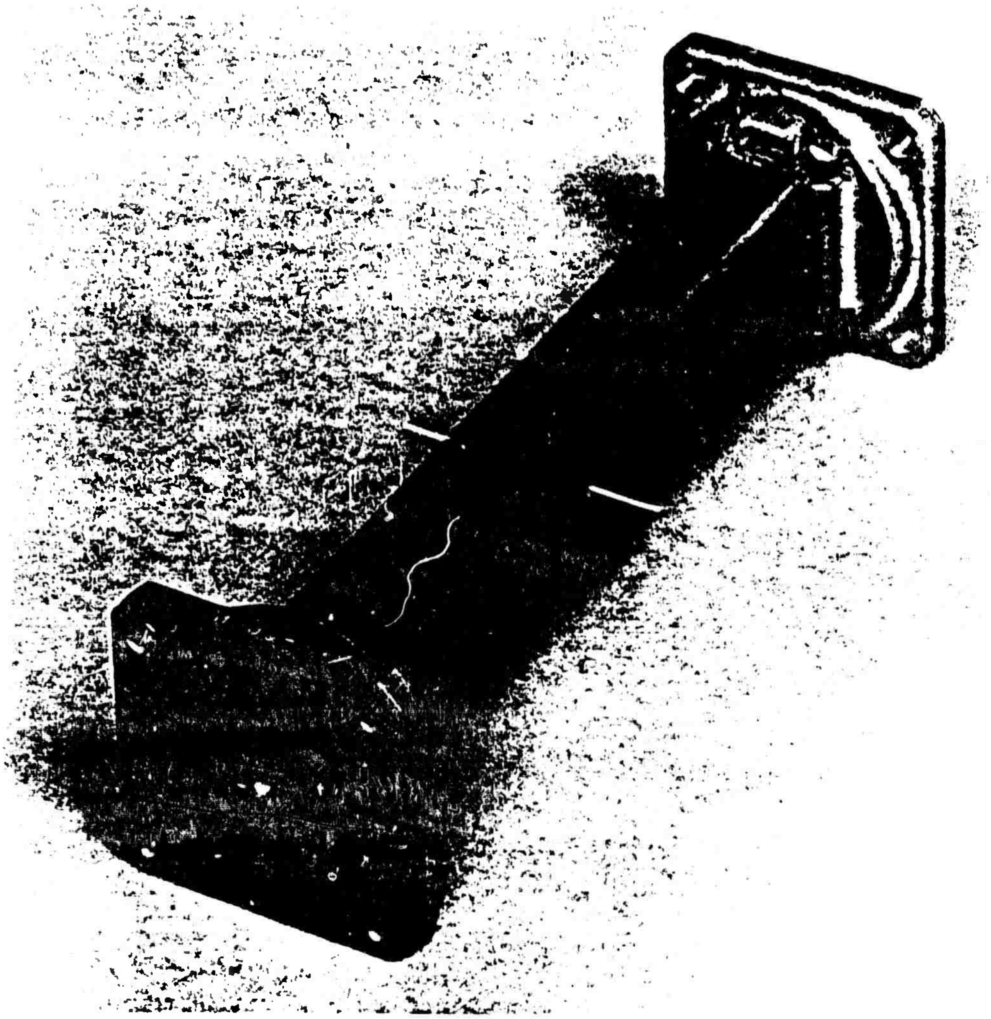


Figure 1. A simple test fixture for the dielectric-post-in-waveguide measurements at X-band.

characteristic impedance of the waveguide Z_0. (See Fig. 2). In this case the equations become:

$$\frac{Z_2}{Z_0} = \frac{-Z_1}{2Z_0} - \frac{aj}{2\lambda_g}\left[\frac{J_0(\beta)}{J_0(\alpha)} \frac{1}{\beta J_0(\alpha)\, J_1(\beta) - J_0(\beta)\, J_1(\alpha)} - S_0 + \frac{\alpha^2}{4}\right]$$

$$\frac{Z_1}{Z_0} = j\frac{2a}{\lambda_g}\left(\frac{\pi d}{a}\right)^2 / \left[\frac{\alpha^2 J_1(\beta)}{J_1(\alpha)} \left(1/(\alpha J_1(\alpha)\, J_1(\beta) - \beta J_0(\beta)\, J_1(\alpha)\right) -2)\right]$$

$$S_0 = \ln\left(\frac{4a}{\pi d}\right) - 2 + \sum_{n=3}^{\infty}{}' \left[\frac{1}{n^2 - 2a} - \frac{1}{n}\right], \qquad \alpha = \frac{\pi d}{\lambda},\ \beta = \epsilon \frac{\pi d}{\lambda}$$

where a is the waveguide width and d is the sample diameter λ is the free space wavelength and λ_g is the wavelength in waveguide. The computer calculation uses the full expression since some samples with large dielectric constants are not electrically thin. The sum is taken to 100 terms for odd n. The complex J_0 amd J_1 Bessel functions are evaluated by series approximations (6). The S-parameters are given by:

$$S_{11} = S_{22} = [(Z_1 + Z_2)^2 - Z_2^2 \quad -1]/[(Z_2 + Z_1 + 1)^2 - Z_2^2],$$

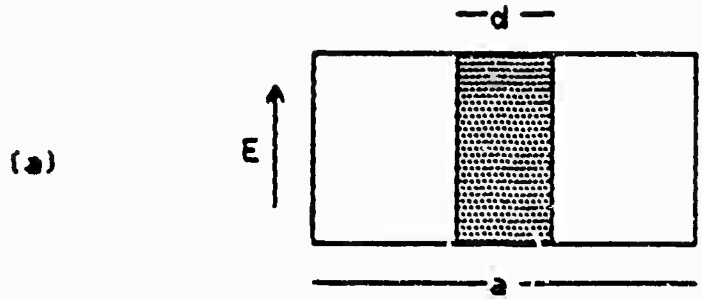
and

$$S_{12} = S_{21} = 2Z_2/[Z_2^2 + 2Z_2Z_1 + 2(Z_2 + Z_1) + 1].,$$

where standard waveguide theory was used to derive the measured circuit parameters from the equivalent circuit representation (7).

EXPERIMENTAL SET-UPS

DPW measurements have been greatly simplified with the advent of accurate automated vector network analyzers. These devices can rapidly scan the waveguide band measuring the S-parameters of the DPW waveguide fixture at hundreds of frequencies. The X(7.5 to 12.4 GHz) and Ku (11 to 18 GHz) data were taken with a Hewlett-Packard Model 8409S and recently with a Model 8510 automatic vector network analyzers. For measurements with the 8409S AVNA we find it convenient to terminate the waveguide test fixture with a good matched load and employ the S11 parameter for the computer inversion. Using just the S11 values speeds up the measurement time, especially saving time in the initial calibration of the AVNA. We find it necessary to place a pad (about 5 dB) before the waveguide fixture to damp out spurious standing waves. Nevertheless, the principal source of error arises from uncorrected standing waves in the measurement arm. With the acquisition of the 8510 AVNA the post



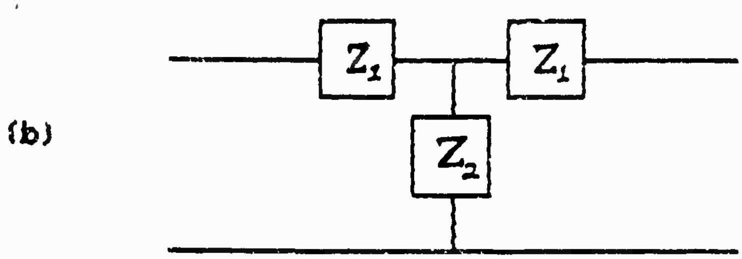


Figure 2. (a) A sketch of the dielectric post positioned in waveguide where a is the width of the waveguide and d is the sample diameter; and, (b) a sketch of the equivalent circuit.

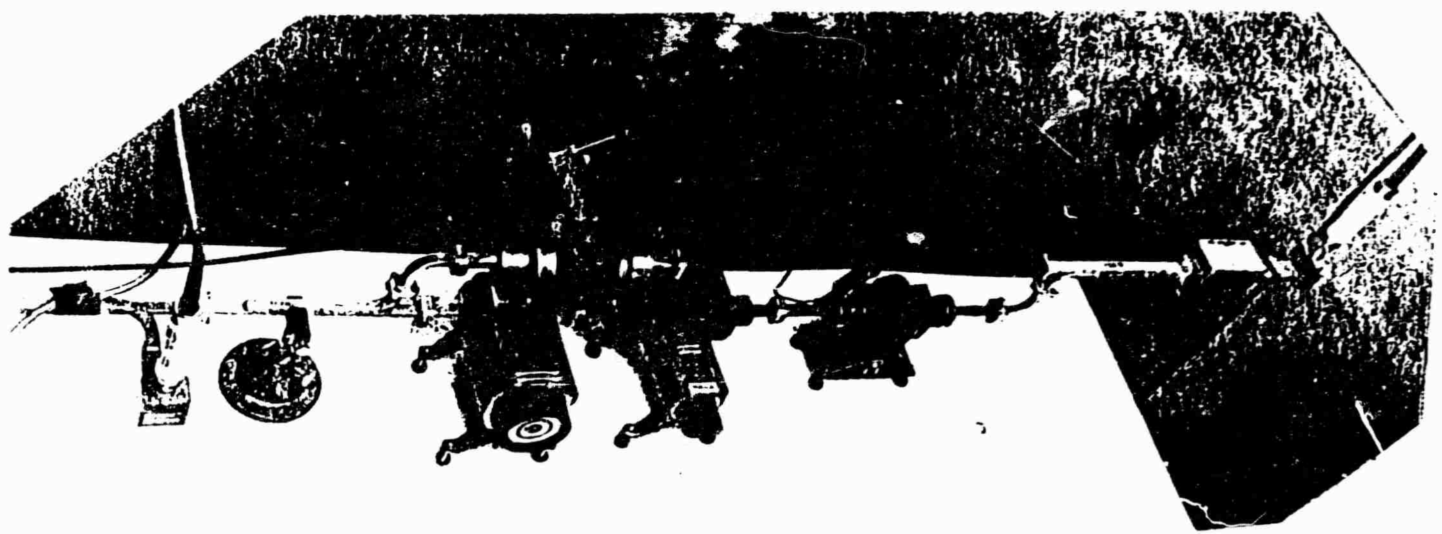


Figure 3. A picture of the Ka band (26.5 to 40 GHz) transmission bridge. The sample is held in the top arm. Leads are pictured attached to the sample in the waveguide for the imposition of a voltage across a filamentary ferroelectric sample. After the microwave sweeper a directional coupler diverts a reference amplitude signal to the scaler network analyser. Precision calibrated attenuators are located in both arms of the bridge with a calibrated phase shifter in the lower (reference) arm.

measurements are much improved owing to an order of magnitude improvement in the accuracy of the complex S-parameter determinations. With the 8510 we measure both S_{11} and S_{21} with small standing wave distortions employing simple unpadded waveguide fixtures. The permittivity can then be infered from S_{11}, S_{21} or S_{11} and S_{21} . The choice of data type is determined by the relative reflection and transmission amplitudes. Least squares fitting of the data employes both S_{11} and S_{21}.

We also have made measurements at K(18 - 26.5 GHz) and Ka band (26.5-40 GHz). Our AVNAs do not cover these waveguide bands and so we have assembled simple transmission bridge circuits at these frequencies. (See Fig. 3). The source in this case is a HP Model 8690B Microwave Sweeper with appropriate BWO plug-ins. An amplitude reference signal is tapped off just after the sweeper and then the microwave power is split into a sample arm and a reference arm. Both sample and reference arms contain precision calibrated attenuators and the sample arm holds the DPW test fixture whereas the reference arm contains a calibrated phase shifter. Microwaves from the two arms are brought together and the mixed signal is detected by a standard wave guide Shotkey detector. Measurements are acquired by a Model 1038-N10 Pacific Measurements Scalar Network Analyzer and fed to a LSI-11 computer for storage on floppy disk. Data is taken as the sweeper is swept across the full waveguide band. The insertion loss is first measured with the reference arm set at maximum attenuation. Phase data is then acquired with the sample in place and with an empty test fixture. For the phase measurement the power in the two arms is adjusted so that the power through the reference arm is slightly greater than that through the sample arm. The phase shifter is then stepped through 360 degrees at (usually) 20 degree intervals. At each phase setting a full sweep is acquired and stored on the LSI/11. The sample arm attenuator is usually set at 7 dB to damp out spurious standing waves and an isolator is used before the mixer-detector for the same purpose. The phase data is then analyzed with a computer code, PHASE, which performs at least squares fit to data at all the phase shifter settings at each acquisition frequency (or a specified subset of frequencies). This fitted phase shift is normalized to the fitted phase shift for data taken with an empty sample holder. In some cases return loss is also measured at a second detector mounted on a directional coupler in the sample arm and referenced to a short inserted just before the waveguide test fixture.

As mentioned earlier the waveguide fixture consists of a simple short length of waveguide with the sample inserted through a hole centered in the broad face. The sample is usually housed in a thin-walled glass capillary tube commonly employed in Debye-Scherer X-ray measurements. Insertion of the test sample by means of a capillary tube through the hole in the wave guide allows rapid sample insertion and for a minimum disturbance of the test network. This sample mounting also allows the imposition of a D.C. voltage across

the sample while it is being measured at microwave frequencies. This sample insertion method may cause problems in the case of high or moderately conducting samples or for high dielectric samples due to depolarization effects at the waveguide walls. For such samples it is probably necessary to make good electrical contact to the waveguide. One method for doing this is to use a waveguide cover flange with a small grove in one face to hold the sample. The sample is cemented in place in the grove with conducting paste (3). On a few occasions we have used a split waveguide fixture to hold larger samples. The samples are cut to form right circular cylinders of very nearly the waveguide height and are then press-fitted into the split section of waveguide for measurement.

COMPUTER CODES

A number of computer codes have been developed to simulate and analyze the data generated by the dielectric post in waveguide experiments. The major codes and their functions are described below. Since the experimental setups are simple, most of the effort in creating the measurement capability was devoted to developing the computer codes. Likewise, the data acquisition is simpler than the data analysis, especially when the data is taken with the automated vector network analyzer. All the codes described in this report are written in Fortran IV or Fortran 77.

Program POSTCL calculates the return or insertion loss and phase for a given simple diameter, waveguide size and frequency scan and for a given dielectric constant and conductivity. Frequency dependent permittivities can be generated by other codes and entered in POSTCL by a data file. This program is used to estimate the experimental data from a known sample (Fig. 4). For example, if a carbon fiber sample of a given diameter and known D.C. conductivity is to be studied, we can calculate if there will be enough sensitivity to detect it or if it will cause such a large impedance mismatch so as to be impractical to analyze. One can then pre-select samples, sometimes bundling several lengths of material before the actual experiments are attempted.

Program EPOST takes return/insertion loss and phase data and matches it by using Newton's method to iterate the equations at each frequency. The program requires an initial guess for the permittivity. One has the option of starting out the program with a single guess for the first frequency and allowing the program to procede to subsequent frequencies starting from the initial guess or starting from the solution for the previous frequency. One also has the option of specifying the initial guess at each frequency by means of an input data file. Such a file is typically generated from an analysis of a similar sample or by extrapolation from the frequency

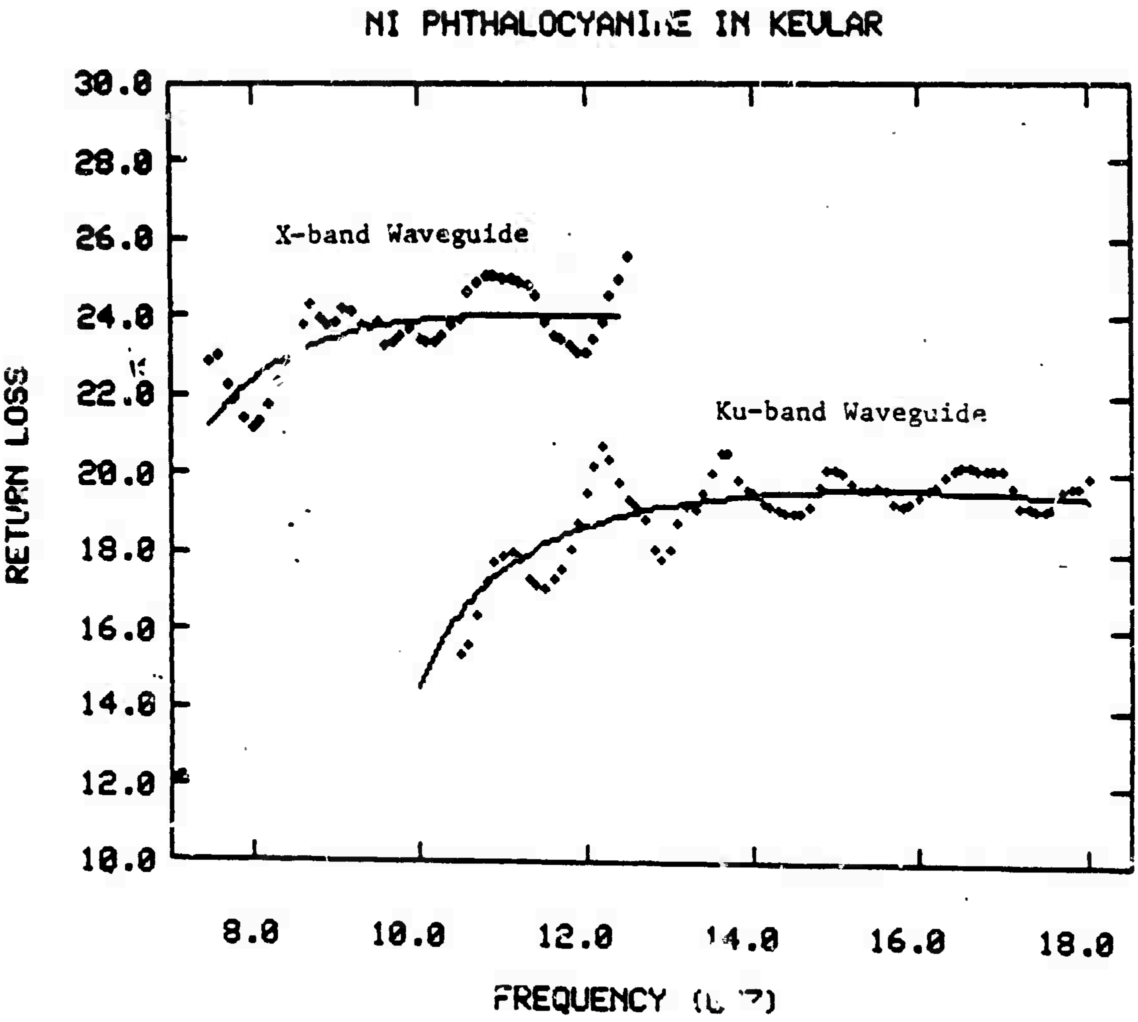


Figure 4. A plot of the measured return loss (+) and the calculated return loss for a sample of Ni substituted phthalocyanine in kevlar. The return loss measured at the two waveguide bands differ since the characteristic impedance of the waveguides are different. The calculated curve is generated by program POSTCL using a complex dielectric constant and complex conductivity found by program POSTTEK.

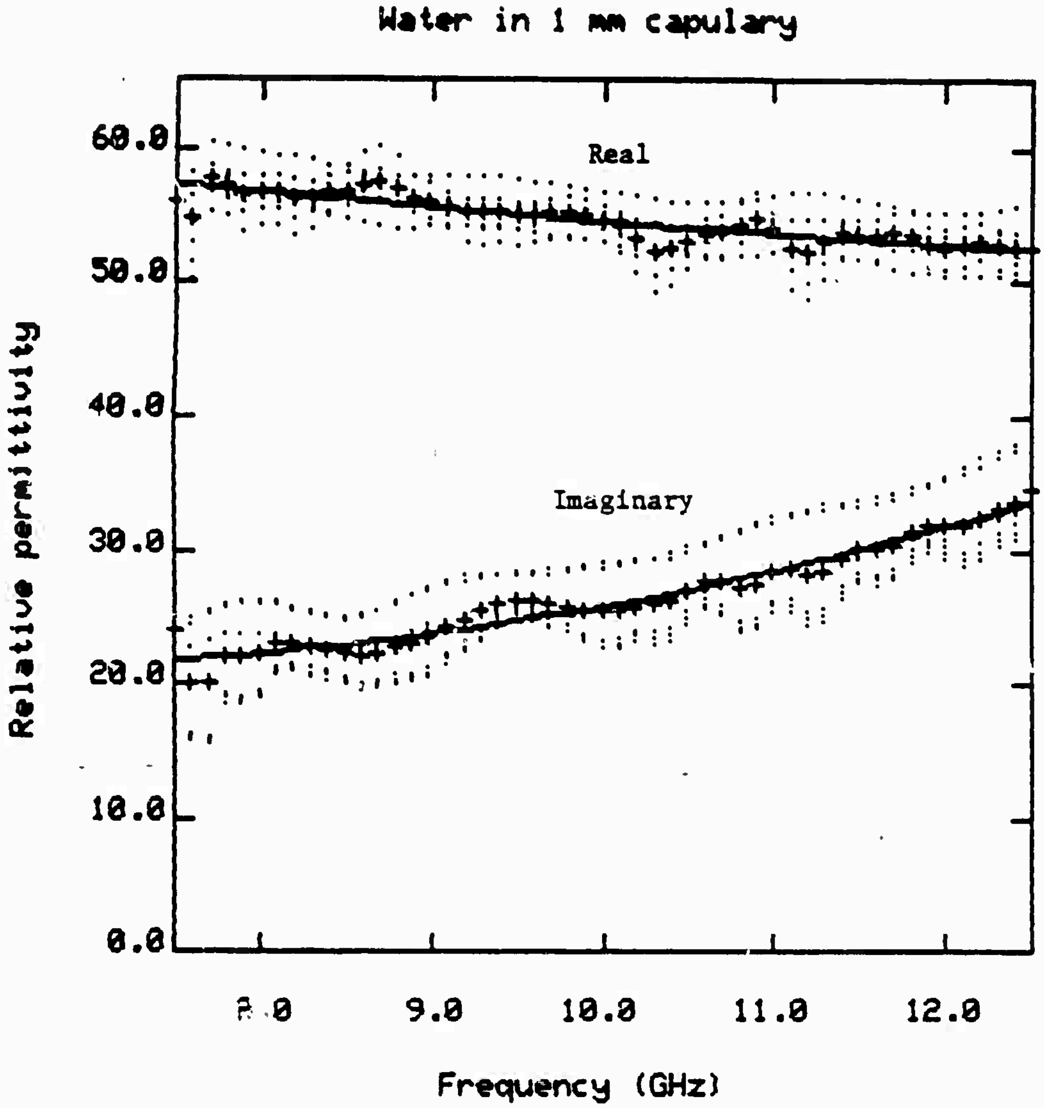


Figure 5. A plot of the relative permittivity of water. The data were taken at X-band using the HP8510 Vector Network analyser and the measurement was repeated three times with the sample repositioned between measurements. The dots are EPOST inversions of the data for both S11 and S21. The crosses result from a REFPOSTTEK analysis on the three sets of data. The line was generated by program Posttek operating on all three data sets and fitting the data to second order polynomial representatives for the frequency dependence of ε' and ε".

dependent fit of the sample at a different waveguide band. The program generates plots (Fig. 5) and a permittivity file which can be used with other programs.

Program REFPOSTTEK performs a least square fit to S-parameter data at each frequency to find ϵ. It differs from EPOST in that a least square fit is used instead of a Newton inversion of the data. The least square approach is less likely to diverge at a bad initial guess for ϵ. Since both the real and imaginary parts of ϵ are found, this program requires at least three data points at each frequency. It thus differs from EPOST which requires exactly two data points for each Newton-Rolfson inversion. The two couples S-parameters S_{11} and S_{21} result in four data points at each frequency and thus suffice for the REFPOSTTEK analysis. Often samples with large permittivities and non-cylndrical crossections or slightly tapering diameters will produce differing S-parameters each time they are inserted in the waveguide fixture. Program REFPOSTEK will "average" over the scatter in measured data. REFPOSTEK is similar to POSTTEK (described below) in that it performs a least square of the data but it differs in that the inversion is performed at each frequency. It does not require a specification of a functional from for ϵ with frequency. The input data is taken from a single waveguide band measurement series.

Program POSTTEK takes complex reflection and/or transmission data at one or more waveguide bands and performs a least-squares best fit of the data to a menu of selected complex functions of frequency representing the frequency dependence of the permittivity. The program will vary one or more of the parameters appropriate to the designated function and print out the optimized values and plot the fit to the data (Fig. 6). The optiizaton of the parmeters is performed by standard least squares subroutines[8]. The menu includes a variety of functions including polynomials, a conductivity function, a damped harmonic oscillator function, a Lorentzian, a Gaussian, etc. The input files are prepared from the AVNA S-parameter files or from dB and phase files by subsidiary codes (NANFIL and XYDBPHS2V). A dielectric function file may be created using the parameters and other subsidiary programs such as PERFIL and TRANSXY. The permittivity file may serve for a first guess for EPOST in analyzing new data or for comparison to the output of EPOST.

Program FTEMTEK is similar to POSTTEK in that it will fit a multitude of data to a specified function of frequency. This program fits the complex permittivity calculated by EPOST or REFPOSTTEK rather than the measured transmission and reflection coefficients. A plot of the fit generated by FTEMTEK is shown in Fig. 7. It is found that the POSTTEK and FTEMTEK programs often result in very nearly the same answers but the FTEMTEK code requires much less computer time since it does not have to repeatedly calculate the post impedance

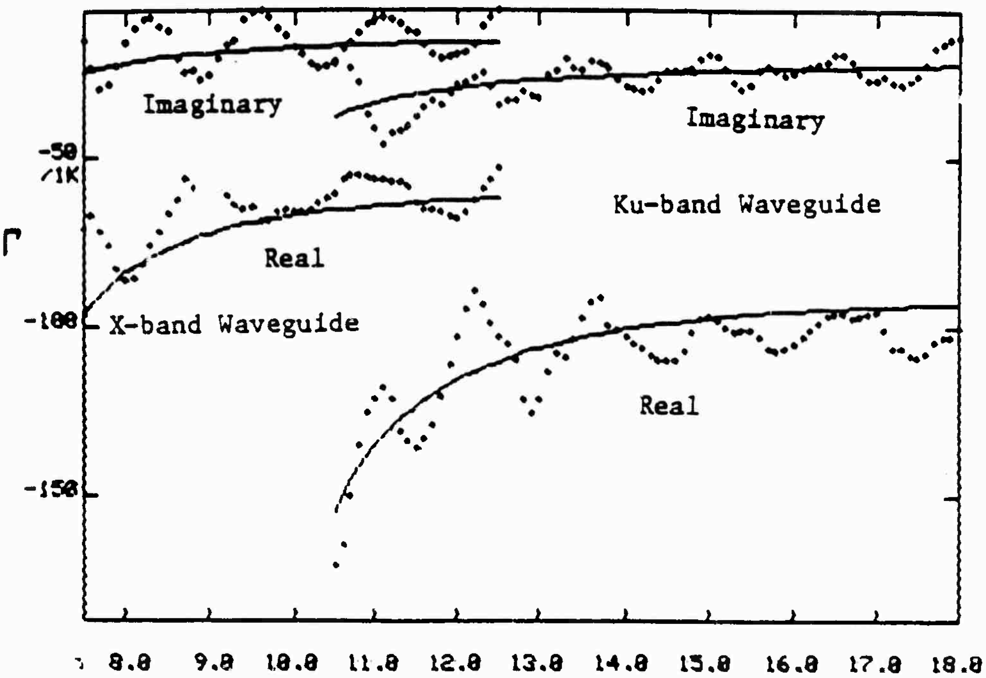


Figure 6. A plot generated by program POSTTEK showing the real and imaginary reflection (X and Ku bands) and the fit for a conducting fiber sample (Ni substituted phthalocyanine in Kevlar).

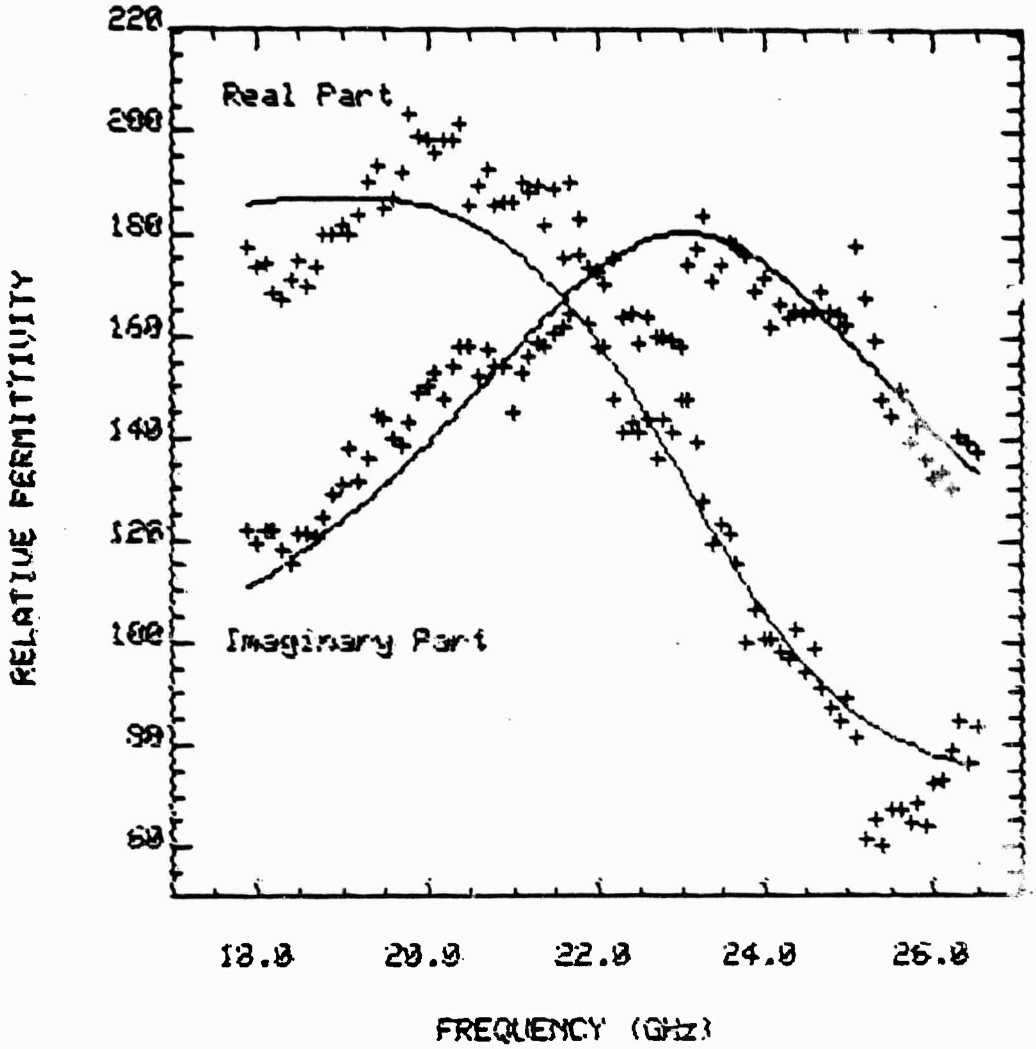


Figure 7. A plot generated by program FTEMTEK showing a fit of dielectric data generated by EPOST. The sample is a ferroelectric compound and the fit (line) is a damped harmonic oscillator function.

equations. Since the input to FTEMTEK is one step removed from actual measured data and relys on the proper convergence of EPOST or REFPOSTTEK, POSTTEK is the prefered code in the case of a difficult data analysis.

As mentioned above, program PHASETEK calculates a phase from the phase data acquired using the transmission bridge. The bridge puts out a number of dB insertion loss files versus frequency for various settings of the calibrated phase shifter in the reference arm of the bridge circuit. (See Fig. 8a). PHASETEK transforms the insertion loss data (dB) into scalar transmission coefficient data and then performs at least squares fit of the data to the equation A + B*sin(C) at each frequency (Fig. 8b). The phase C(f) are saved on a file and the difference between the phase with the sample present is subtracted from the phase without the sample using program ASMDXY. The phase difference file is used for the evaluation of the dielectric response.

Codes POSTCLTEK, POSTTEK, REFPOSTTEK and EPOST rely on a subroutine package POST which calculates the equivalent impedances and the complex reflection and transmission coefficients. This subroutine package is listed in the Appendix along with the relevant subroutines and calling programs of some of the previously described codes.

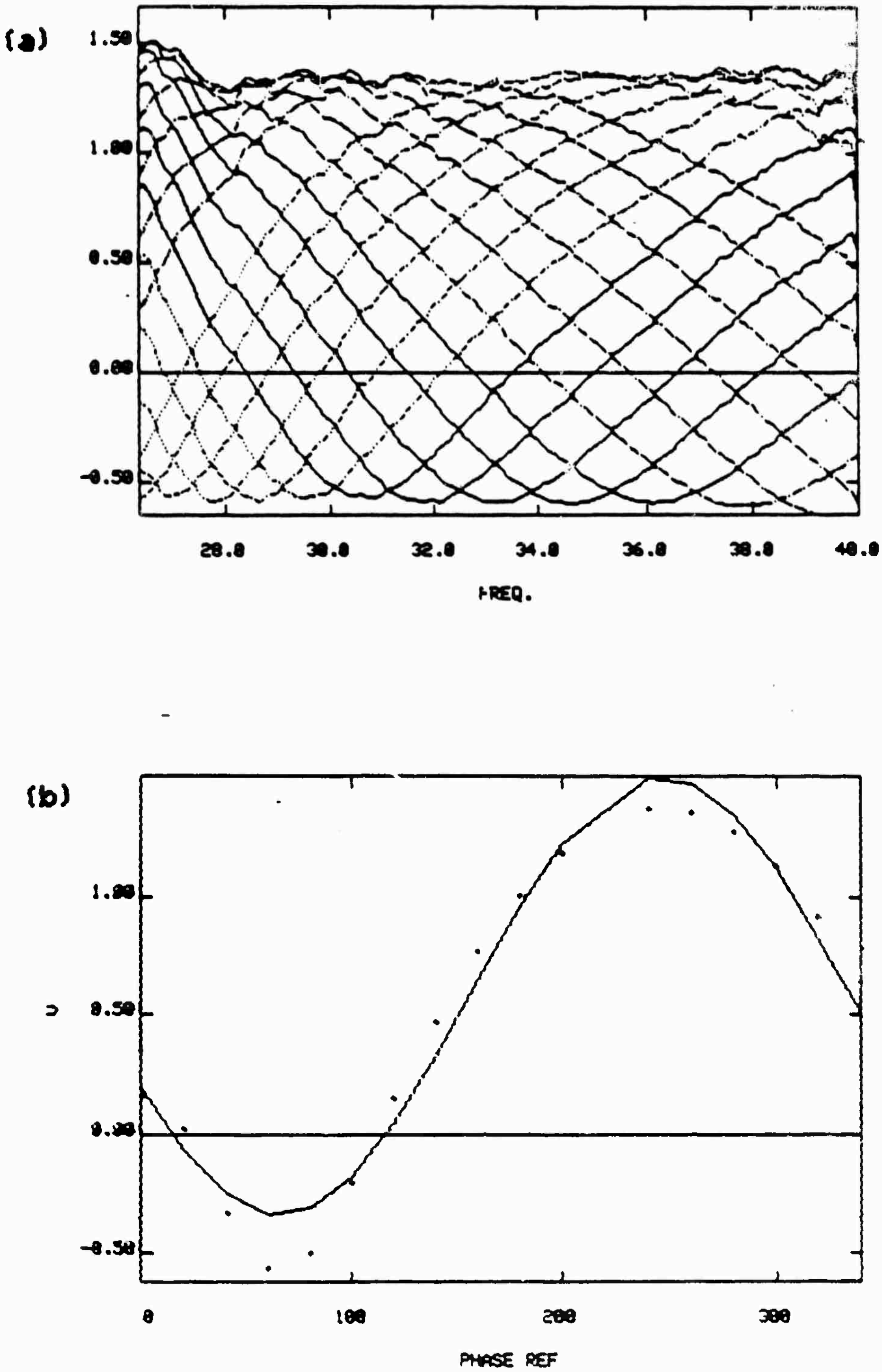


Figure 8. Plots generated by program PHASETEK (a) showing all the input phase data versus frequency and (b) a single frequency fit of the phase data versus phase shifter setting at a single frequency. In figure 8a there is one curve for each 20 degrees of phase shift. The plots are displayed in differing color and line style on a color terminal to help distinguish the various curves.

EXAMPLES

We examine the use of the above programs in the analysis of several samples. The first sample examined is a partially graphitized carbon fiber designated DG-129-R. A single fiber has a diameter of the order of one micrometer and produces a negligible impedance change in the waveguide. A bundle of the order of 50 fibers was needed to produce a reasonable signal. We estimate the sample diameter of the fiber bundle by measuring the length and weight of the bundle and calculating the effective diameter from the known density of the material. The plots of the return loss data produced by the AVNA at X and Ku bands are shown in Fig. 9. The periodic oscillations are due to the presence of standing waves which could not be completely cancelled or corrected for in the error correction software of the model 8409S AVNA. The insertion loss and the phase difference produced by the transmission bridge after the use of program PHASE and subtraction of the phase output from the empty holder are shown in Fig. 10. The Ka band bridge data appear to be of higher quality tnan the AVNA data but are more difficult to acquire than that using the automated system. Standing-wave damping appears to be more complete.

In the next figure (Fig. 11) we display the output files of EPOST for all three waveguide bands. The standing-wave oscillations noted in the AVNA data are transformed into oscillations in the apparent permittivity with respect to frequency. A clear trend is seen in the data consistent with an inverse frequency falloff of the imaginary part of the dielectric constant as is expected from a simple conductor. A small frequency dependence is seen in the real part of ϵ. The solid lines in the figure are the frequency dependent permittivity given by program FTEMEK assuming a frequency independent ϵ coupled with a complex conductivity: σ = 0.78645-.00022j and ϵ = 2.886-0.3242j. The imaginary part of the conductivity is assumed responsible for the small fall-off in the real part of the permittivity with increasing frequency. the coefficients calculated for a similar functional fit to the reflection and transmission data via program POSTTEK are very nearly the same within a few percent. Since the X and Ka band data are rather noisy we have also fit just the Ka band data by itself to a frequency indepedent complex ϵ and a real conductivity using POSTTEK. This fit results in the following coefficients: ϵ = 13.20 - 187j and a real conductivity σ =2.560/Ohm-cm. The conductivity is within 12 of that inferred from the larger set of data.

As a second example we study the case of a polycrystaline ferroelectric sample. A sheet of the material was polished to a thickness of 12 mil and then sliced to produce needles with a rectangular 12 x 12 mil crosssection. A needle was selected and measured at X and Ka bands employing the Model 8510 AVNA. Although the calculations assume cylindrical symmetry it is found that

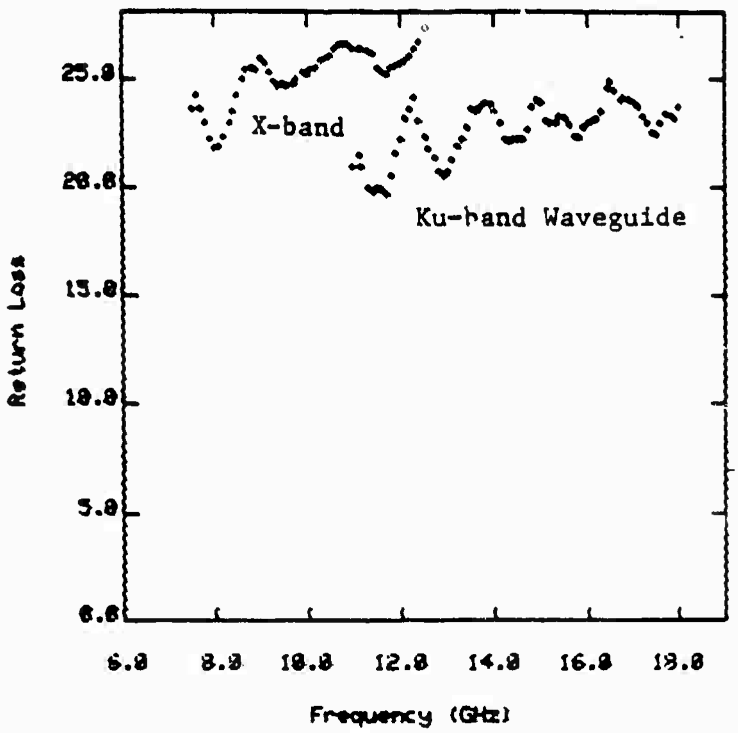


Figure 9. A plot of the return loss of a conducting fiber bundle taken by the automated vector network analyser at X and Ku bands.

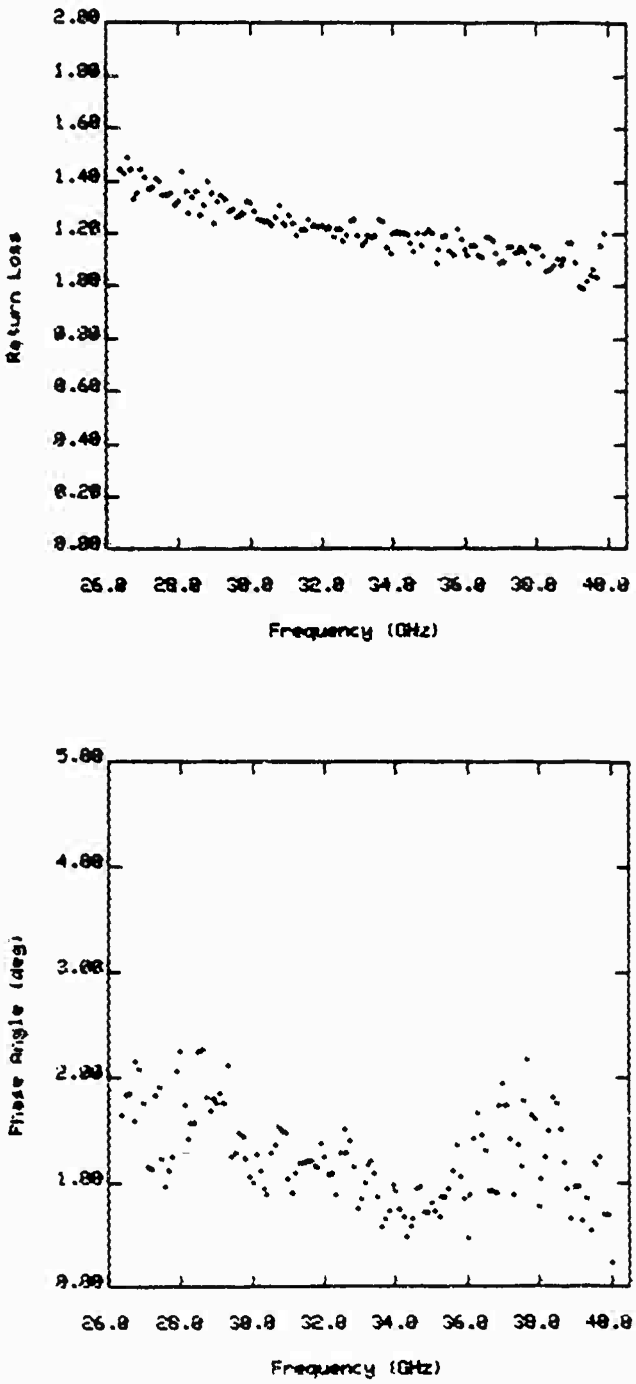


Figure 10. A plot of the insertion loss and phase difference data taken on the transmission bridge at Ka band for conducting fiber bundle (DG-129-R).

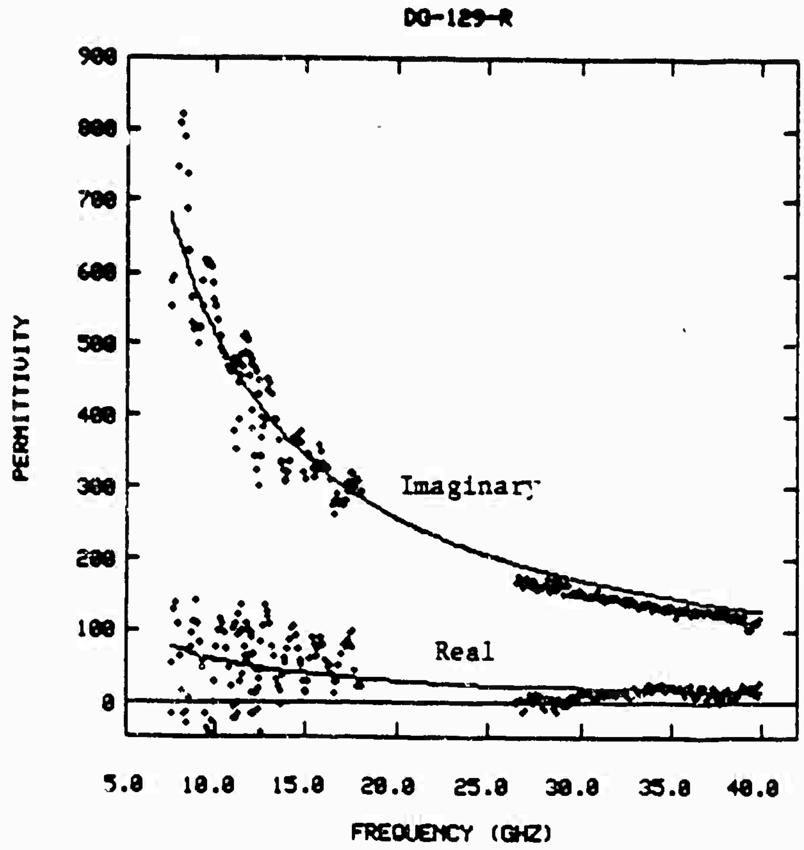


Figure 11. The EPOST dielectric points along with fits to the data (lines) for fiber bundle sample DG-129-R. The data was taken at three different waveguide bands; X, Ku, and Ka. The best fit curve was generated using parameters from FTEM or POSTTEK (nearly identical numbers) for all three bands.

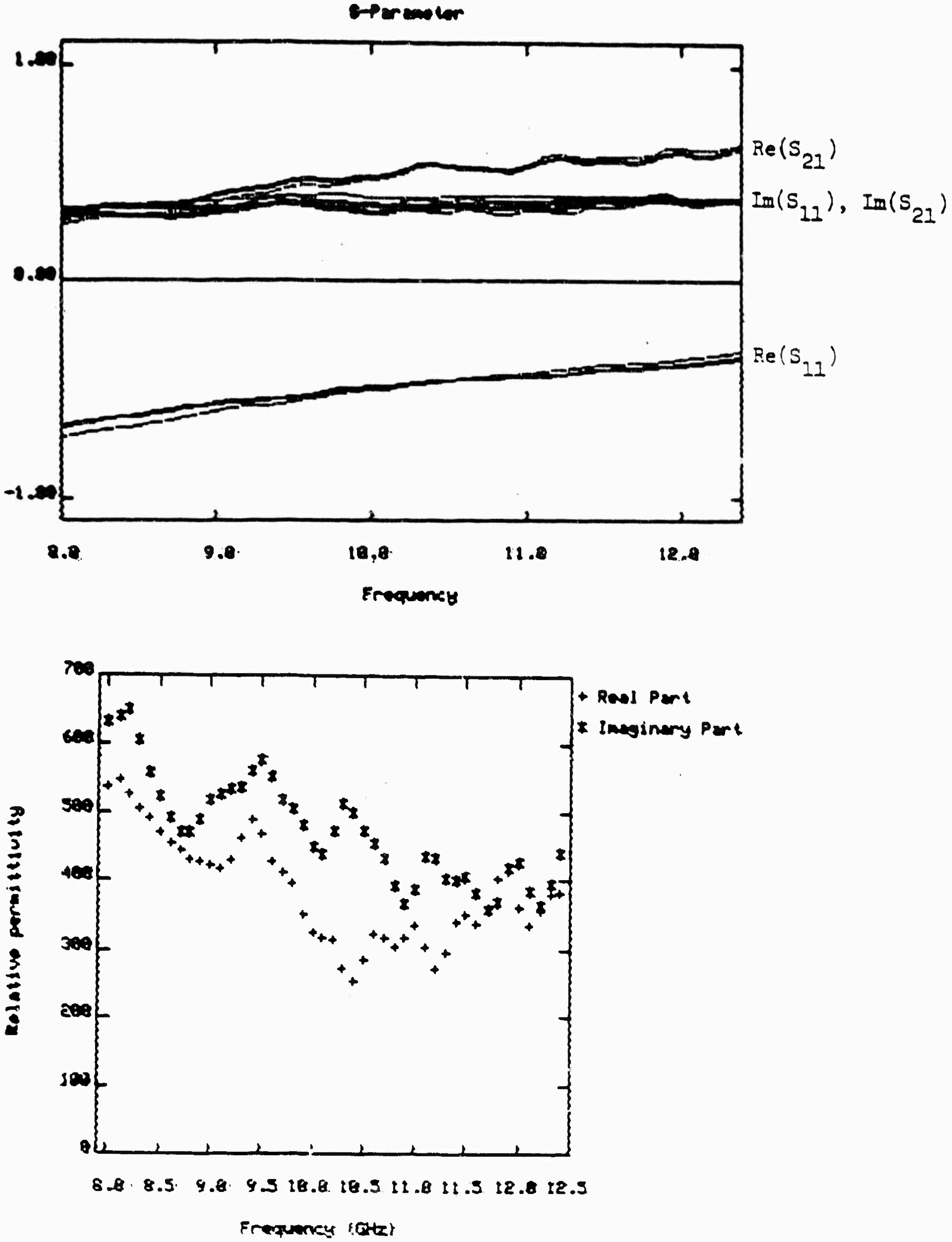


Figure 12. $Re(S_{11})$, $Im(S_{11})$, $Re(S_{21})$ and $Im(S_{21})$ data (points) and Computer fits (lines) for polycrystaline ferroelectric needles measured X and Ka bands using a HP Model 8510 AVNA and fit with G code FTPOST.

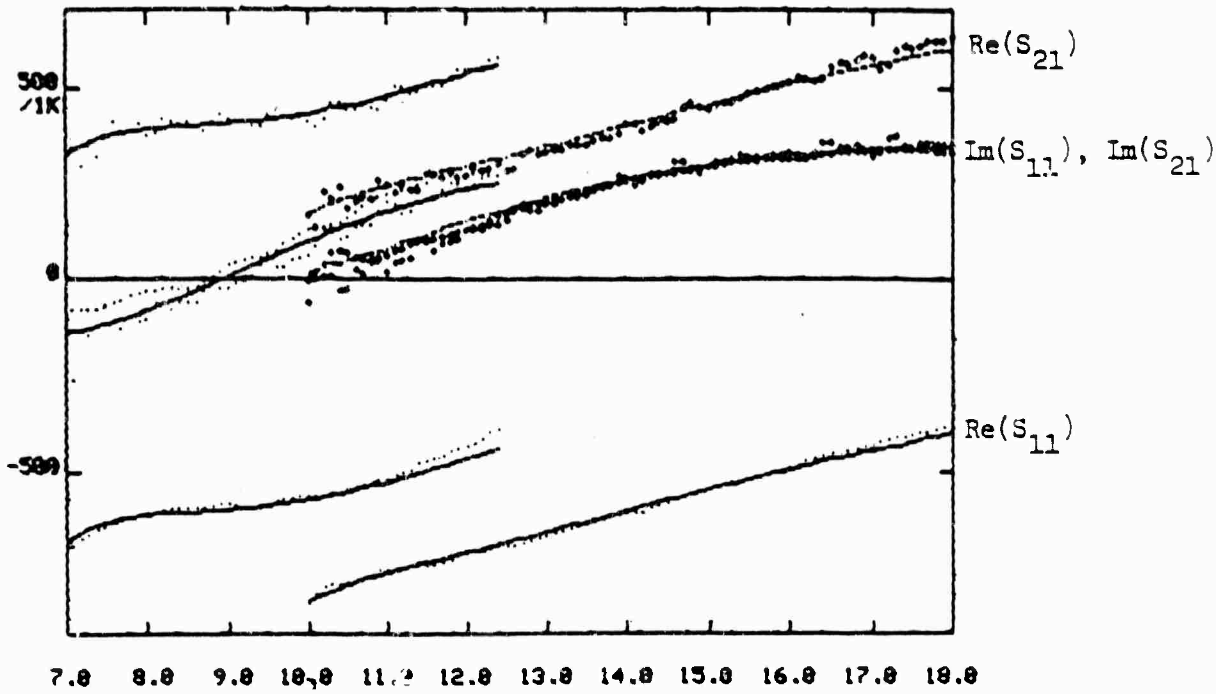


Figure 13. a) Input S_{11} and S_{21} data for three independent measurements of a ferroelectric sample are plotted as well as b) the permittivity of the sample as fitted at each frequency by program REFPOSTTEK.

reasonable results can be obtained for samples with small rectangular crossections where an effective sample diameter is calculated assuming a circular crossection of equal area. No waveguide attenuators were used to suppress standing waves. The S parameter data shows much less scatter than encountered in the previous example at the same frequency bands. (See Figure 12). The data was fit using POSTTEK and assuming a quadratic frequency dependence, $\varepsilon^* = A + Bf + Cf^2) - j(D + Ef + Ff^2)$, over the frequency range studied, where f is the frequency in GHz. A slightly different ferroelectric formulation was studied at X-band using the HP8510 Network Analyzer. Three independent measurements of the X-band S-parameters were analyzed using program REFPOSTTEK. The twelve input curves (Re and Im S_{11} and S_{21}) are shown in Figure 13 along with the inverted complex permittivity.

CONCLUSIONS

This report describes the theory, experimental set-ups, and computer codes developed at NRL for the use of the dielectric post-in-waveguide technique for the measurement of dielectric materials. The technique is especially useful in studying the properties of filamentary high loss dielectric materials. The method is mostly used for room temperature studies though some work has been done warming the sample holder and reference line to 40C and cooling to 10C. We expect to be able to extend this limited temperature range to -100 to +300C. The experimental technique is simple; the data analysis is also strightforeward but the method requires substantial computer computations.

ACKNOWLEDGEMENTS

I would like to thank Dr. Charles Young, Mr. Joseph Peak and Mr. Scott Browning for taking the data discussed in this paper and for help in the construction of the test fixtures. I would also like to thank Drs. Dawn Dominguez and Manfred Kahn for supplying some of the samples.

REFERENCES

(1) Jean Le Bot, Ann Phys, 11,463-492, (56); Translation NIH Library NIH-72-336.

(2) R. Beleckas, J. Grigas, Lietuvos Fiz. Rinkinys (Soviet Physics — Collections), XIX, 243 (1979).

(3) W.J. Gunning, A.J. Heeger, I.F. Shchegolev, S.P. Zolotukhin, Solid State Commun., 25, 981-985 (1978).978).978).978).978).978).978).978).978).

(4) Julian Schwinger and David S. Saxon, DISCONTINUITIES IN WAVEGUIDES, Gordon and Breach New York (1968).

(5) Waveguide Handbook, Franck, R.L. Repost Tg, 1942, sect V.

(6) Handbook of Mathematical Functions with formulas, graphs, and mathematical tables, Edited by Milton Abvamovitz and Irene A. Stegun, National Bureau of Standards, Applied Mathematics Series Government Printing Office No. 55, Washington, DC (1964), pp. 369-370.

(7) R.W. Beatty, D.M. Kerns, Proc IEEE 52, p. 84, 1964.

(8) Data Reduction and Error Analysis for the Physical Sciences, Phillip R. Bevington, McGraw-Hill, New York, p. 235, 1969.

APPENDIX I
Measurement and Analysis Procedure

In this section we list the steps involved in the measurment and data reduction of a sample by our DPW method.

SAMPLE SELECTION AND MOUNTING

A sample is chosen with a large length to diameter ratio to minimize depolarization effects. Usually the sample is inserted in a Debye-Scherer capillary tube with outside diameter small enough to fit through the smallest hole in the waveguide test fixtures (0.5mm). For lower dielectric constant material a post is fashioned for each waveguide band with diameters usually less than a third of the waveguide width and length closely matching the waveguide height. The latter samples are press-fitted into position in split waveguide sections instead of being inserted through a hole in the waveguide. For high conductivity fibers or some filamentary linear chain ferroelectric crystals a fine fiber filament is found which spans the waveguide at the lowest measurement frequency and is cemented across a waveguide cover flange with conducting paste. A small grove is cut in the flange to mount the sample in this mannaer alligning it to span the middle of the waveguide. The sample is cut from the flange fixture with a razor blade and mounted in successively smaller flange mounts for higher frequency measurements.

AUTOMATED VECTOR NETWORK ANALYSER MEASUREMENTS

To use the AVNA it must first be calibrated for measurements at a particular waveguide band. The calibration procedure is described in the AVNA manual and takes approximately 10 minutes. The calibration on the HP8409S AVNA is performed with 5 to 10 dB isolation pad waveguide sections in place. No pads are needed on the HP 8510 AVNA. The test fixture is in place during the transmission segments of the calibration procedure. We usually require the return loss from the empty test fixture after calibration to be 35 dB down or more over the frequency range before the calibration is accepted and measurements begin.

After the calibration, the sample is placed in the test fixture and the S-parameters are measured. The AVNA lists and plots the S-parameters. From the listing and plots generated by the AVNA we determine if the data is adequate for further analysis. If not the sample may be repositioned, a sample with a larger or smaller diameter may be found, or other adjustments may be made to lower the standing wave interference. Once the data is acceptable it is stored on a floppy disk before other samples are measured. Later the Hewlett-Packard floppies are transcribed to IBM exchange format for transport to a VAX computer for analysis.

TRANSMISSION BRIDGE MEASUREMENTS

The transmission bridge is assembled as shown in Fig. 3. The sweeper is turned on set, to sweep in free run across the waveguide band and allowed to stabilize for approximately an hour before measurements begin. Initially the reference arm is turned off by setting the attenuator in that arm to maximum and the signal through the sample arm is used to normalize the scalar network analyser. The normalization is done with an empty capillary tube in the place of the sample when appropriate and the sample arm attenuator is set to 7 dB to damp standing waves. The reference arm attenuator is then set back until the power through the reference arm is slightly larger than that through the sample arm at all frequencies. The phase shifter set to 0 deg; a trace is acquired by the network analyser and transfered to floppy disk on a DEC LSI computer. The phase shifter is advanced and other traces are acquired until data is taken for phase settings from 0 to 340 degrees in steps of 20 degrees.

After the empty bridge has been characterized the sample is inserted in the test fixture. The reference arm is again turned off and the intertion loss is measured on the network analyzer and transfered to disk. The reference arm is then turned on and the power set so that the reference level is once again slightly larger than the power through the sample arm at all frequencies. The phase shifter is set to zero and traces are taken and stored on disk at successive phase shifter settings as done previously. Once the phase data has been acquired, a new sample is inserted and measured. Some bridge setups have a 10dB directional coupler in the sample arm to measure the scalar reflection coefficient of the sample. To make this measurement, a short is inserted at the sample position to normalize the network anlayzer and then the short is removed and the sample is placed in position. The scaler netowork analyzer trace is recorded on floppy disc for later analysis.

COMPUTER ANALYSIS

Programs to analyze the SPW data are run on a VAX 11/780 computer. In the following description of a data reduction we assume that a sample has been measured using both a Hewlett-Packard automated vector network analyser and a transmission bridge.

Data from the AVNA is brought to the PDP 11/34 on floppy disks written in the IBM exchange fromat. The floppy disk is then carried to the VAX. Once on the VAX the data is placed into standard file structures which are callable by a wide vareity of programs. NANFIL takes the ASCII S-parameter file and outputs dB versus frequency and phase versus frequency as unformatted binary "XY" files. (The binary files take up less disk space.) Since the sample is usually

positioned some distance behind the AVNA reference plane when the measurements are performed, a program OFFPHS is invoked to adjust the phase files to account for this offset. DB and phase files are then ready to be run using program EPOST.

Data from the transmission bridge is written on floppy disks as binary "XY" files in the RT-11 format by the LSI-11 computer at the experimental station. The floppies are then carried to the VAX, read and reformatted to VAX VMS files using program RT2VMS. The insertion loss data files are directly usable by our analysis programs, however, the phase data requires considerable manipulation to obtain the phase shift versus frequency required for later analysis. To obtain the phase shift program PHASETEK must be invoked twice; once to find the initial phase versus frequency for data taken with an empty sample fixture, and then, to obtain the pase versus frequency with the sample. Thus PHASETEK is called up and the relevant data files are read in and an initial guess is given for the parameters of the fitting sine function. PHASETEK will then output phase vs frequency binary files. The phase file for measurements with the sample is then subtracted from the file for measurements with an empty sample holder to obtain the relative phase shift versus frequency. The subtraction is done using program ASMDXY. The insertion loss and phase files are then ready for EPOST.

Program EPOST calculates the permittivity of a sample from a dB and a phase value at each frequency. The program internally minimizes the difference between the measured real and imaginary parts of the voltage reflection or transmission coefficient and the calculated value via Newton's method. DB and phase data files generated for a single measurement of a sample in a given waveguide are entered into EPOST along with the sample diameter and waveguide size. The diameter is measured with a micrometer or a calibrated traveling microscope for a single fiber sample and is calculated for fiber bundles from a known density and the measured length and mass. Program EPOST requires a guess for the permittivity at the first frequency in order to start off its itterative search for the permittivity. If this guess is not close enough the interations may diverge. If such divergences are a problem, the dB and phase versus frequency may be calculated for a number of guesses for the dielectric constant and conductivity using program POSTCL and compared to the measured data. Or it is often more useful to take a subset of the input data files (using ASSMBL to form the smaller files) and run POSTTEK with a simple dielectric function since this program is less susceptable to numerical divergences. Once an acceptable initial guess is found, EPOST inverts the data frequency by frequency and generates binary permittivity files. Observation of the EPOST plots will usually suggest trial function for the frequency dependence of the permittivity.

If enough data is available program REFPOSTTEK can be employed instead of EPOST. REFPOSTTEK performs a least squares fit to the S-parameter data at each frequency point. It requires more input data than fitted parameters; thus, to calculate the real and imaginary parts of the permittivity at least three data curves are required. They could be, for instance, the real and imaginary parts of S_{21} along with the magnitude of S_{21}. Though more data are prefered including several independent measurements of the same sample or several samples of the same material with different diameters. To use REFPOSTTEK the dB and phase data are rewritten as real and imaginary S-parameters vs frequency via program XYDBPHSTOV. Amplitude S-parameter files are created from dB data by program DB2VXY. The least square interaction procedure is not as sensitive to the initial guess of the permittivity as is EPOST. The program will accept a permittivity file generated by another program to specify the initial values for ε at each frequency or it will accept an initial guess of the real and imaginary parts of the permittivity at the first frequency point and enquire whether this guess is to be applied at all subsequent points or whether subsequent points are to be initiated at the previous soluation.

Program POSTTEK fits the measured complex reflection or transmission data from one or more measuremnts of the sample to a menu selected function of frequency. In order to invoke POSTTEK, the dB and corrected phase files are combined and written to disk as real and imaginary voltage coefficient versus frequency files via program XYDBPHS2V. POSTTEK will also accept scalar voltage coefficients and will simultaneously analyze data taken at several waveguide bands. PHASETEK optimizes the coefficients of the specifiec functions and can put out files of the calculated voltage coefficients from the optimized functions. The program runs rather slowly due to the lengthy nature of the post calcuations.

Often nearly identical functional dependences are obtained much more rapidly using program FTEMTEK. To use this program permittivity files generated by EPOST or REFPOSTTEK are split into real and imaginary parts using program SPLTFL and entered in FTEMTEK. FTEMTEK then fits the EPOST/REFPOSTTEK data to the same menu specified funtions of frequency and outputs files of the calculated optimized fits. The outputted files can be used by program TID to plot the fit and the data in formats suitable for publication. The output files also can be combined and used as input files for POSTCL calculations or for first guess permittivity files for further EPOST or REFPOSTTEK itterations.

The measurement and calculation procedures are thus seen to require many steps, but once the equipment and computer codes are "up and running" the analysis is essentially straightforward.

APPENDIX II
Program Listings

INTRODUCTION

In this appendix are listed some of the FORTRAN programs used in the DPW analysis. The first listing is the subroutine package POST.FOR which includes the post impedance calculations and the conversion to reflection and transmission parameters. This is followed by the calling programs of POSTCL.FOR and EPOST.FOR. The subroutines under the title FTPOST.FOR are presented. These subroutines are linked to a general fitting program package (FITTEK.FOR) to produce POSTTEK. Only a small number of changes are required in FTPOST to produce FTEM.FOR which is linked to the fitting routines to produce FTEMTEK. The difference of files FTPOST.FOR and FTEM FOR is listed in the last part of this appendix.

THE POST CALCUATION: POST.FOR

The following subroutine package calculates the post equivalent circuit impendances and the reflection and transmission parameters for a dielectric post centered in a rectangular waveguide.

```
      SUBROUTINE POST(EZ1,EZ2,A,D,F,EPEP)
C
C
C      CALCULATES THE EQUIVALENT IMPEDANCES OF A DIELECTRIC POST
C
C                      IN WAVEGUIDE
C
C
C                     ----Z1-----Z1----
C                               !
C                               Z2
C                               !
C                     ----------------
C
C
C
      IMPLICIT COMPLEX (E)
      EP = CONJG(EPEP)
      PIE = 3.1415927
      ALPH =  0.10471976*F*D
      EBET = ALPH*CSQRT(EP)
      EJOB = EBESJO(EBET)
      EJ1B = EBESJ1(EBET)
      CALL BESJO(ALPH,JO,AJOA,V,IERR)
      CALL BESJ1(ALPH,JO,AJ1A,V,IERR)
      ALAM = 30./F
      ALAMG = 2*A*ALAM/SQRT(-ALAM*ALAM + 4*A*A)
D     TYPE *,F,A,D,AJOA,AJ1A,EJOB,EJ1B
      A1 = A/(2.*ALAMG)
      A2 = PIE*D/A
      A2SQ = A2*A2
      EQ1 = A1*((EJOB/(AJOA*(EBET*AJOA*EJ1B-ALPH*EJOB*AJ1A)))
     1-SO(A,D,ALAM)+ALPH*ALPH*0.25)
      EQ2 = 4*A1*A2SQ/((ALPH*ALPH*EJ1B/(AJ1A*(ALPH*AJOA*EJ1B
     1-EBET*EJOB*AJ1A)))-2.)
      EI = CMPLX(0.,1.)
      EZ2 = -EI*(0.5*EQ2+EQ1)
      EZ1 = EI*EQ2
      RETURN
      END
      FUNCTION SO(A,D,ALAM)
      ABL=2.*A/ALAM
      SO = ALOG(1.2732395*A/D)-2.
      DO 1 J = 1,100
      AJODD = 2*J+1
      ACUM = ((1./SQRT(AJODD*AJODD-ABL*ABL))-1./AJODD)
1     CONTINUE
      SO = SO + 2.*ACUM
      RETURN
      END
```

```
      FUNCTION E1(EZ,T,N)
C
C     USED WITH EBESJ TO CALCULATE COMPLEX BESSEL FCT
C
      IMPLICIT COMPLEX(E)
      ST = SIN(T)
      EST = EZ*ST-N*T
      EI = CMPLX(0.,1.)
      EST = EI*EST
      EEST = CEXP(EST)
      E1 = 0.5*(EEST+1./EEST)
      E1 = E1/3.1415927
      RETURN
      END
      FUNCTION F1(T)
      IMPLICIT BYTE (Q)
      IMPLICIT COMPLEX(E)
      COMMON /VARIB/EZ,N,QREAL,Q
      IF(.NOT.QREAL)GO TO 1
      F1 = REAL(E1(EZ,T,N))
      RETURN
1     F1 = AIMAG(E1(EZ,T,N))
      RETURN
      END
```

```
      FUNCTION ETPOST(A,D,F,EP)
C
C     CALCULATES THE COMPLEX TRANSMISSION COEF
C
C     S12  FOR A POST IN A RECTANGULAR WAVEGUIDE
C
C
      IMPLICIT COMPLEX (E)
C
C
C
      CALL POST(EZ1,EZ2,A,D,F,EP)
D     TYPE *,'Z1,Z2',EZ1,EZ2
C     ETPOST = EZ2/(EZ1*EZ1 + EZ1 + EZ2 + 2.*EZ1*EZ2)
      ETPOST = 2.*EZ2/(EZ1*EZ1 + 2.*(EZ1*EZ2+EZ1+EZ2)+1.)
      RETURN
      END
```

```
      FUNCTION ERPOST(A,D,F,EP)
C
C
C     CALCULATES THE COMPLEX REFLECTION COEF
C
C     S11  --  FOR A DIELECTRIC POST IN
C
C     RECTANGULAR WAVEGUIDE
C
C
      IMPLICIT COMPLEX(E)
      CALL POST(EZ1,EZ2,A,D,F,EP)
      E1 = EZ1 + EZ2
      E12 = E1*E1
      E22 = EZ2*EZ2
      E3 = E1 + 1.
      E32 = E3*E3
      ERPOST = (E12 - 1. - E22)/(E32 - E22)
      RETURN
      END
```

```
      SUBROUTINE TPOSTD(TDB,TPH,A,D,F,EP)
C
C
C     CALCULATES THE INSERTION LOSS AND PHASE
C
C     FROM A DIELECTRIC POST IN MATCHED RECTANGULAR WAVEGUIDE
C
C
      IMPLICIT COMPLEX (E)
      E = ETPOST(A,D,F,EP)
D     TYPE *,'T',E
      TDB = -20.*ALOG10(CABS(E))
      TPH = 57.29578*AIMAG(CLOG(E))
      RETURN
      END
```

```
      SUBROUTINE RPOSTD(RDB,RPH,A,D,F,EP)
C
C
C     CALCULATES THE RETURN LOSS AND PHASE
C
C     FROM A DIELECTRIC POST IN MATCHED RECTANGULAR WAVEGUIDE
C
C
      IMPLICIT COMPLEX (E)
      E = ERPOST(A,D,F,EP)
      RDB = -20.*ALOG10(CABS(E))
      RPH = 57.29578*AIMAG(CLOG(E))
      RETURN
      END
```

```
      FUNCTION REPOST(EP,A,D,F,QT)
C
C
C     CALCULATES THE REAL PART OF THE
C
C     REFLECTION OR TRANSMISSION COEFFIEIENTS
C
C
      IMPLICIT BYTE (Q)
      IMPLICIT COMPLEX(E)
      IF(QT)GO TO 1
      REPOST = REAL(ERPOST(A,D,F,EP))
      GO TO 2
1     REPOST = REAL(ETPOST(A,D,F,EP))
2     RETURN
      END
```

```
      FUNCTION AIPOST(EP,A,D,F,GT)
C
C
C     CALCULATES THE IMAGINARY PART OF THE
C
C     REFLECTION OR TRANSMISSION COEFFIEIENTS
C
C
      IMPLICIT BYTE (Q)
      IMPLICIT COMPLEX (E)
      IF (QT) GO TO 1
      AIPOST = AIMAG(ERPOST(A,D,F,EP))
      GO TO 2
1     AIPOST = AIMAG(ETPOST(A,D,F,EP))
2     RETURN
      END
```

```
      FUNCTION ABPOST(EP,A,D,F,QT)
C
C     CALCULATES THE ABSOLUTE VALUE OF THE
C
C     REFLECTION AND TRANSMISSION COEFFICIENTS
C
C
      IMPLICIT BYTE(Q)
      IMPLICIT COMPLEX(E)
      IF (QT) GO TO 1
      ABPOST = CABS(ERPOST(A,D,F,EP))
      GO TO 2
1     ABPOST = CABS(ETPOST(A,D,F,EP))
2     RETURN
      END
C*************************************************************U*******U***
C
C
C     SUBROUTINE EBESJO
C
C
C     COMPLEX JO BESSEL FUNCTION FROM A  POLYNOMIAL APPROXIMATION
C
C     TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS..
C     US DEPT OF COMMERCE  -- NBS  APPLIED MATH SERIES 55
C     EQS 9.4.4 AND 9 4 6
C
C
C     GOOD FOR RANGE -3 TO 3 ERROR LESS THAN 1.3E-8
C     AND FROM 3 TO INFINITY WITH ERROR LESS THAN 4E-9
C
C     IN PARAENTHESIS LIKE PDP ROUTINE EXCEPT JO AND E DO NOTHING
C     SX IS COMPLEX*8
C     SBJ IS COMPLEX*8
C     IER IS ERROR CONDITION
C
C
C8**********************************************************************
      SUBROUTINE EEBESJO(SX,JO,SBJ,E,IER)
      IMPLICIT DOUBLE PRECISION (A,B,T,F,X)
C     REAL A(18)
      COMPLEX*8 SBJ,SX
      COMPLEX*16 BJ,BBJ,X3BY,X,XBY3,TH,FO,THO
      DOUBLE PRECISION A(18)
      DATA A/-2.2499997D0,1.2656208D0,-.3163866D0,.0444479D0
     1,-.0039444D0,.0002100
     2,-.00000077D0,-.0055274D0,-.00009512D0,.00137237D0,
     3-.00072805D0,.00014476D0
     4,-.04166397D0,-.00003954D0,.00262573D0,-.00054125D0
     5,-.00029333D0,.00013558D0/
      X = SX
      XBY3 = X/3.
      IF (REAL(X).LT.-3 )IER = 2
      IF (ABS(X).GT.3) GO TO 1
      BBJ =DCMPLX(1.D0,0.D0)
      DO 2 J = 1,6
      J2 = 2.*J
2     BBJ = BBJ + A(J)*XBY3**J2
      BJ = BBJ
      GO TO 3
```

```
1       X3BY = 1./XBY3
        BBJ = DCMPLX(0.79788456D0,0.D0)
        TH = DCMPLX(-.78539816D0,0.D0)
        DO 4 J = 1,6
        JP = J + 6
        JPP = J + 12
        BBJ = BBJ + A(JP)*X3BY**J
4       TH = TH + A(JPP)*X3BY**J
        FO = BBJ
        THO = X + TH
        BJ = FO*CDCOS(THO)/CDSQRT(X)
3       SBJ = CMPLX(BJ)
        RETURN
        END
C*************************************************************************
C
C
C       SUBROUTINE BESJ1
C
C
C       COMPLEX J1 BESSEL FUNCTION FROM A  POLYNOMIAL APPROXIMATION
C
C       TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS.
C       US DEPT OF COMMERCE  -- NBS  APPLIED MATH SERIES 55
C       EQS 9.4.4 AND 9.4.6 ,
C
C
C       GOOD FOR RANGE -3 TO 3 ERROR LESS THAN 1.3E-8
C       AND FROM 3 TO INFINITY WITH ERROR LESS THAN 4E-8
C
C       IN PARAENTHESIS LIKE PDP ROUTINE EXCEPT J1 AND E DO NOTHING
C       SX IS COMPLEX*8
C       SBJ IS COMPLEX*8
C       IER IS ERROR CONDITION
C
C
C8***********************************************************************
        SUBROUTINE EEBESJ1(SX,J1,SBJ,E,IER)
        IMPLICIT DOUBLE PRECISION (A,B,T,F,X)
        COMPLEX*8 SX,SBJ
        COMPLEX*16 X,XBY3,X3BY,BBJ,BJ,TH,F1,TH1
C       REAL A(20)
        DOUBLE PRECISION A(20)
        DATA A/-.56249985D0,.21093573D0,-.03954289D0,.00443319D0,
        1-.00031761D0,.00001109D0,.79788456D0,.00000156D0,.01659667D0,
        2.00017105D0,-.00249511D0,.00113653D0,-.00020033D0,
        3-2.35619449D0,
        4.12499612D0,.0000565D0,-.00637879D0,.00074348D0,.00079824D0,
        5-.00029166D0/
        X = DCMPLX(SX)
        XBY3 = X/3.
        IF (REAL(X).LT.-3.)IER = 2
        IF (CDABS(X).GT.3) GO TO 1
        BBJ = DCMPLX(0.D0,0.D0)
        DO 2 J = 1,6
        J2 = 2.*J
2       BBJ = BBJ + A(J)*XBY3**J2
        BJ = X*(.5 + BBJ)
        GO TO 3
1       X3BY = 1./XBY3
```

```
      BBJ = DCMPLX(0.D0,0.D0)
      TH = DCMPLX(0.D0,0.D0)
      DO 4 J = 1,6
      JP = J + 7
      JPP = J + 14
      BBJ = BBJ + A(JP)*X3BY**J
4     TH = TH + A(JPP)*X3BY**J
      F1 = A(7) + BBJ
      TH1 = X + A(14) + TH
      BJ = F1*CDCOS(TH1)/CDSQRT(X)
3     SBJ = BJ
      RETURN
      END
C*************************************************************U*******U***
C
C
C     SUBROUTINE BESJO
C
C
C     JO BESSEL FUNCTION FROM A  POLYNOMIAL APPROXIMATION
C
C     TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS.
C     US DEPT OF COMMERCE  -- NBS  APPLIED MATH SERIES 55
C     EQS 9.4.4 AND 9.4.6
C
C
C     GOOD FOR RANGE -3 TO 3 ERROR LESS THAN 1.3E-8
C     AND FROM 3 TO INFINITY WITH ERROR LESS THAN 4E-9
C
C     IN PARAENTHESIS LIKE PDP ROUTINE EXCEPT JO AND E DO NOTHING
C     SX IS SINGLE PRECICION INPUT
C     SBJ IS SINGLE PRECISION OUTPUT
C     IER IS ERROR CONDITION
C
C
C8**************************************************************************
      SUBROUTINE BESJO(SX,JO,SBJ,E,IER)
      IMPLICIT DOUBLE PRECISION (A,B,T,F,X)
C     REAL A(18)
      DOUBLE PRECISION A(18)
      DATA A/-2.2499997D0,1.2656208D0,-.3163866D0,.0444479D0
     1,-.0039444D0,.00021D0
     2,-.00000077D0,-.0055274D0,-.00009512D0,.0013723700,
     3-.00072805D0,.00014476D0
     4,-.04166397D0,-.00003954D0,.00262573D0,-.00054125D0
     5,-.00029333D0,.00013558D0/
      X = SX
      XBY3 = X/3.
      IF (X.LT.-3.)IER = 2
      IF (ABS(X).GT.3) GO TO 1
      BBJ = 1.
      DO 2 J = 1,6
      J2 = 2.*J
2     BBJ = BBJ + A(J)*XBY3**J2
      BJ = BBJ
      GO TO 3
1     X3BY = 1./XBY3
      BBJ = 0.79788456D0
      TH = -.78539816D0
      DO 4 J = 1,6
```

```
      JP = J + 6
      JPP = J + 12
      BBJ = BBJ + A(JP)*X3BY**J
4     TH = TH + A(JPP)*X3BY**J
      FO = BBJ
      THO = X + TH
      BJ = FO*DCOS(THO)/DSQRT(X)
3     SBJ = BJ
      RETURN
      END
C**************************************************************
C
C
C     SUBROUTINE BESJ1
C
C
C     J1 BESSEL FUNCTION FROM A  POLYNOMIAL APPROXIMATION
C
C     TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS
C     US DEPT OF COMMERCE  -- NBS  APPLIED MATH SERIES 55
C     EQS 9.4.4 AND 9.4.6
C
C
C     GOOD FOR RANGE -3 TO 3 ERROR LESS THAN 1.3E-8
C     AND FROM 3 TO INFINITY WITH ERROR LESS THAN 4E-8
C
C     IN PARAENTHESIS LIKE PDP ROUTINE EXCEPT J1 AND E DO NOTHING
C     SX IS SINGLE PRECICION INPUT
C     SBJ IS SINGLE PRECISION OUTPUT
C     IER IS ERROR CONDITION
C
C
C8*****************************************************************
      SUBROUTINE BESJ1(SX,J1,SBJ,E,IER)
      IMPLICIT DOUBLE PRECISION (A,B,T,F,X)
C     REAL A(20)
      DOUBLE PRECISION A(20)
      DATA A/-.56249985D0,.21093573D0,-.03954289D0,.00443319D0,
     1-.00031761D0,.00001109D0,.79788456D0,.00000156D0,.01659667D0,
     2.00017105D0,-.00249511D0,.00113653D0,-.00020033D0,
     3-2.35619449D0,
     4.12499612D0,.00005650D0,-.00637879D0,.00074348D0,.00079824D0,
     5-.00029166D0/
      X = SX
      XBY3 = X/3.
      IF (X.LT.-3.)IER = 2
      IF (ABS(X).GT.3) GO TO 1
      BBJ = 0.
      DO 2 J = 1,6
      J2 = 2.*J
2     BBJ = BBJ + A(J)*XBY3**J2
      BJ = X*(.5 + BBJ)
      GO TO 3
1     X3BY = 1./XBY3
      BBJ = 0
      TH = 0.
      DO 4 J = 1,6
      JP = J + 7
      JPP = J + 14
      BBJ = BBJ + A(JP)*X3BY**J
```

```
4       TH = TH + A(JPP)*X3BY**J
        F1 = A(7) + BBJ
        TH1 = X + A(14) + TH
        BJ = F1*DCOS(TH1)/DSQRT(X)
3       SBJ = BJ
        RETURN
        END
        SUBROUTINE SIMUL(II,VAR,OK,QOUT)
        IMPLICIT BYTE(Q)
        DIMENSION VAR(II)
        DIMENSION DVAR(2),EP(2)
        QOUT = .FALSE.
D       TYPE *,'SIMUL'
        DO 1 I = 1,II
        DVAR(I) = VAR(I)*.0001
        IF(DVAR(I).EQ.0.)DVAR(I) = 1.E-5
1       CONTINUE
C       TYPE *,'VAR =',VAR,'DVAR =',DVAR
D       TYPE *,'VAR =',VAR,'DVAR =',DVAR
2       CALL CORREC(EP,II,VAR,DVAR)
D       TYPE *,'EP =',EP
        EPSIL = 0.
        DO 3 I = 1,II
        E1 = 0.
        IF(EP(I).EQ.0.OR.VAR(I).EQ.0)GO TO 3
        E1 = ABS(EP(I)/VAR(I))
3       EPSIL = EPSIL + E1
C       IF (DMAX.LT.ABS(VAR(1))) QOUT = .TRUE.
C       IF (EPSIL.LT.OK.OR.QOUT) GO TO 4
        DO 5 I = 1,II
5       VAR(I) = VAR(I) + EP(I)
D       TYPE *,'VAR(I) =',VAR
        IF (EPSIL.LT.OK)GO TO 4
        GO TO 2
4       RETURN
        END
```

```
      SUBROUTINE CORREC(EP,II,VAR,DVAR)
      DIMENSION VAR(II),DVAR(II),EP(2),F(2),DF(2,2)
D     TYPE *,'CORREC'
      DO 1 I = 1,2
      EP(I) = 0.
1     F(I) = FUNI(I,VAR)
      CALL FDERVV(2,VAR,DVAR,DF)
      DENOM = DF(1,1)*DF(2,2)-DF(1,2)*DF(2,1)
      IF (DENOM.EQ.0) GO TO 10
      EP(1) = (F(2)*DF(1,2)-F(1)*DF(2,2))/DENOM
      EP(2) = (F(1)*DF(2,1)-F(2)*DF(1,1))/DENOM
10    CONTINUE
D     TYPE *,'F1,F2,EPSIL1,EPSIL2',F,EP
      RETURN
      END
```

```
      SUBROUTINE FDERVV(II,VAR,DVAR,DF)
      DIMENSION VAR(II),DVAR(II),DF(II,II)
D     TYPE *,'FDERVV'
      DO 2 I = 1,II
      DO 1 J = 1,II
      V = VAR(J)
      D = DVAR(J)
D     TYPE *,'FDERVV V,D =',V,' ',D
      VAR(J) = V+D
      F = FUNI(I,VAR)
D     TYPE *,'F=',F
      VAR(J) = V - D
      DF(I,J) = (F-FUNI(I,VAR))/(2.*D)
1     VAR(J) = V
2     CONTINUE
D     TYPE *,'FDERVV DF(,) =',DF
      RETURN
      END
```

```
      FUNCTION FUNI(I,VAR)
      IMPLICIT BYTE (Q)
      IMPLICIT COMPLEX(E)
      DIMENSION VAR(2)
      COMMON/VALUE/VAL(2),A,D,F,QTRANS,QTANDR
D     TYPE *,'FUNI'
      EP = CMPLX(VAR(1),VAR(2))
      IF(QTANDR)GO TO 6
      GO TO (1,2)I
1     FUNI = REPOST(EP,A,D,F,QTRANS) - VAL(1)
      GO TO 3
2     FUNI = AIPOST(EP,A,D,F,QTRANS) - VAL(2)
      GO TO 3
6     GO TO (4,5)I
4     FUNI = ABPOST(EP,A,D,F,.FALSE.)-VAL(1)
      GO TO 3
5     FUNI = ABPOST(EP,A,D,F,.TRUE.)-VAL(2)
3     RETURN
      END
```

THE CALCULATED RESPONSE FROM KNOWN SAMPLE: POSTCL FOR

This program listing calculates the reflection and transmission loss and phase for a sample with a known dielectric constant and conductivity. The permittivity may be specified over the waveguide frequency range using a binary file prepaired by another code.

```
      PROGRAM POSTCL
      IMPLICIT BYTE (Q)
      IMPLICIT COMPLEX(E)
      DIMENSION X(201),Y(201),Z(201)
      CHARACTER*25 QNAM,QNAME,QPLTDS,QNAMD,QNAMP
C
C
C
      TYPE 11
      PRINT 11
11    FORMAT(' CALCUALTES THE INSERTION OR RETURN LOSS & PHASE OF',/
     1' A DIELECTRIC POST IN WAVEGUIDE')
C
C
C
C     INCLUDE 'PLTDES.FOR'
C
C
C
      QPLT = QUER('PLOTS? (TEK 4105)',17)
      IF (QPLT)CALL PLOTS

      QTRANS = QUER('TRANSMISSION CALCULATION?',25)
      PRINT *
      IF(QTRANS)PRINT *,'TRANSMISSION CALCULATION'
      IF(.NOT.QTRANS)PRINT *,'REFLECTION CALCULATION'
      PRINT *
      TYPE *,'WIDTH OF GUIDE (MIL)'
      ACCEPT *,WIDE
      PRINT *,'WIDTH OF GUIDE =',WIDE,'MIL'
      A = 2.54E-3*WIDE
      TYPE * ,'DIAMETER OF POST (MIL)'
      ACCEPT *,DIAM
      PRINT *,'DIAMETER OF POST =',DIAM,'MIL'
      D = 2.54E-3*DIAM

C             YY FILE IN?

      QYY = QUER('YY FILE INPUT?',15)
      IF(QYY)THEN
              TYPE *,'ENTER YY FILE NAME?'
              CALL FYINN(QNAME,NPT,NPTS,FL,FH,DF,X,Y,Z,0)
              PRINT 10,QNAME
              GO TO 13
      END IF
10    FORMAT(' INPUT PERMITTIVITY FROM FILE: ',A25)


      TYPE *,'FREQUENCY RANGE: FLO, FHI, DILTA F (GHZ)'
      ACCEPT *,FL,FH,DF
13    PRINT *,'FREQUENCIES FROM',FL,'TO',FH,'IN STEPS OF',DF,'GHZ'
      NSTEP = (FH-FL)/DF + 1

      IF (QYY) GO TO 12

      TYPE *,'DIELECTRIC CONSTANT ='
      ACCEPT *,REP,AIEP
      PRINT *,'DIELECTRIC CONSTANT =',REP,AIEP

      COND = 0.
```

```
        GCOND = QUER('CONDUCTIVITY?',13)
        IF(GCOND)THEN
                TYPE *,'CONDUCTIVITY IN /OHM-CM ='
                ACCEPT *,COND
                PRINT *,'CONDICTIVITY =',COND,'/OHM-CM'
        END IF

12      PRINT *
        DO 1 J = 1,NSTEP

        IF (QYY) THEN
                F = X(J)
                EP = CMPLX(Y(J),Z(J))
        ELSE
                F = FL + (J-1)*DF
                AIMEC = AIEP + 1800.*COND/F
                EP = CMPLX(REP,AIMEC)
                X(J)=F
        END IF

        IF(QTRANS)GO TO 20
        CALL RPOSTD(Y(J),Z(J),A,D,F,EP)
        GO TO 21
20      CALL TPOSTD(Y(J),Z(J),A,D,F,EP)
21      CONTINUE
1       PRINT *,J,X(J),Y(J),Z(J)
C
C       PLOT DB AND PHASE
C
        CALL PLTSTF(NSTEP,X,Y,1,4,'DI POST',' FREQ.  ','IN LOSS')
C       CALL PLTSTF(NSTEP,X,Z,1,0)
        CALL WNTCPY
        CALL PLTSTF(NSTEP,X,Z,1,5,'DI POST',' FREQ.  ',' PHASE ')
        CALL WNTCPY
        QSAV = QUER('SAVE ON FILES?',14)
        IF(.NOT.QSAV)GO TO 2
        PRINT *
        IF(QTRANS)TYPE *,'INSERTION LOSS FILE'
        IF(.NOT.QTRANS)TYPE *,'RETURN LOSS FILE'
        CALL XYOUT(NSTEP,QNAMD,A,D,0.,X,Y,0)
        PRINT 14, QNAMD
14      FORMAT(' DB FILE',X,A25)
        IF(QTRANS)TYPE *,'INSERTION PHASE FILE'
        IF(.NOT.QTRANS)TYPE *,'RETURN PHASE FILE'
        CALL XYOUT(NSTEP,QNAMP,A,D,0.,X,Z,0)
        PRINT 15,QNAMP
15      FORMAT(' PHASE FILE',X,A25)
2       CONTINUE
        STOP
        END
```

PERMITTIVITY VERSUS FREQUENCY FROM A SINGLE MEASUREMENT: EPOST FOR

EPOST uses dB and phase data files from a single DPW measurement along with the sample diameter and waveguide width to itteratively calculate the permittivity frequency by frequency. The initial guess for the permittivity at each frequency is supplied from a single complex permittivity typed in a run time, from the permittivity calculated at the immediately prededing frequency, or from a file generated by another code.

```
      PROGRAM EPOST
C
C
C
C     PROGRAM EVALUATES THE DIELECTRIC CONSTANT
C
C     OF A POST INSERTED IN THE CENTER
C
C     OF MATCHED RECTANGULAR WAVEGUIDE
C
C     GIVEN THE INSERTION OR RETURN LOSS
C
C     IN DB AND THE ASSOCIATED PHASE SHIFT IN DEG
C
C     OR THE SCALAR RETURN AND INSERTION LOSS IN DB
C
C
C
      IMPLICIT BYTE (Q)
      IMPLICIT COMPLEX(E)
      DIMENSION F(501),Y(501),Z(501),VAR(2),VARIG(2)
     1,FST(501),YST(501),ZST(501)
      COMMON /VALUE/VAL(2),A,D,FF,QTRANS,QTANDR
      CHARACTER*25 QDB,QNAME,QPHASE,QNAMEI
C
C
C
      CALL PRDAT
C
C     INPUT DATA TYPE
C
11    FORMAT(' INSERTION LOSS (DB) XY-FILE INPUT')
12    FORMAT(' RETURN LOSS (DB) XY-FILE INPUT')
13    FORMAT(' ASSOCIATED PHASE (DEG) XY-FILE INPUT')
14    FORMAT(' THE FILES DO NOT HAVE THE SAME NUMBER OF PTS! ')
2     QTANDR = .FALSE.
      QTRANS = QUER('TRANSMISSION MEASUREMENT?',25)
      IF(.NOT.QTRANS)QTANDR= QUER('SCALAR T AND R INPUT?',21)
C
C     READ DATA FILES
C
      IF(QTANDR)GO TO 4
      IF(.NOT.QTRANS)GO TO 7
C
C     TRANSMISSION DB AND PHASE FILES
C
8     PRINT *,'PROGRAM EPOST --
     1SOLVES DIEL CONST FROM POST TRANS DB AND PHASE'
      TYPE 11
      CALL XYIN(NPTS,QDB,A,B,C,F,Y,0)
      TYPE 13
      CALL XYIN(NPTSS,QPHASE,A,B,C,F,Z,0)
      IF(NPTS.EQ.NPTSS)GO TO 5
      TYPE 14
      GO TO 8
C
C     REFLECTION DB AND PHASE FILES
C
7     PRINT*,'PROGRAM EPOST --
     1SOLVES DIEL CONST FROM POST REFL DB AND PHASE'
```

```
      TYPE 12
      CALL XYIN(NPTS,QDB,A,B,C,F,Y,0)
      TYPE 13
      CALL XYIN(NPTSS,QPHASE,A,B,C,F,Z,0)
      IF(NPTS.EQ.NPTSS)GO TO 5
      TYPE 14
      GO TO 7
C
C     REFLECTION AND TRANSMISSION DB FILES
C
4     PRINT *,'PROGRAM EPOST --
     1SOLVES DIEL CONST FROM POST SCALAR REFL AND TRANS'
      TYPE 12
      CALL XYIN(NPTS,QDB,A,B,C,F,Y,0)
      TYPE 11
      CALL XYIN(NPTSS,QPHASE,A,B,C,F,Z,0)
      IF (NPTS.EQ.NPTSS)GO TO 9
      TYPE 14
      GO TO 4
9     PRINT *
      PRINT 15,QDB,QPHASE
15    FORMAT (' INPUT FILE NAMES: RETURN LOSS FILE=',X,A25,',
     1INSERTION LOSS FILE=',X,A25)
      GO TO 6
5     PRINT *
      PRINT 10,QDB,QPHASE
10    FORMAT(' INPUT FILE NAMES: DB FILE=',X,A25,',
     1PHASE FILE=',X,A25)
6     PRINT *
C
C     WAVEGUIDE AND SAMPLE PARAMETERS
C
      TYPE *,'WIDTH OF WAVEGUIDE'
      ACCEPT *,WGW
      PRINT *,'WAVEGUIDE WIDTH=',WGW,'MIL'
      TYPE *,'DIAMETER OF SAMPLE'
      ACCEPT *,DIAS
      PRINT *,'DIAMETER OF SAMPLE=',DIAS,'MIL'
18    FORMAT(//,'    STEP',5X,'FREQ',10X,'FILE  1',7X,'FILE  2',8X,
     1'RE(EP)',9X,'IM(EP)',//)
C
C     CONVERT TO CM FROM MIL
C
      D = 2.54E-3*DIAS
      A = 2.54E-3*WGW
C
C     INITIAL GUESS OF DIELECTRIC CONSTANT AT LOWEST FREQ
C
      QLAST=.FALSE.
      QFIL=QUER('START FROM YY FILE INPUT?',25)
      IF(QFIL)THEN
      CALL FYINN(QNAMEI,N,NPT,F1,F2,DF,FST,YST,ZST,0)
      PRINT 22,QNAMEI
      TYPE 22,QNAMEI
22    FORMAT(' STARTING VALUES FROM FILE ',X,A25)
      VAR(1)=YST(1)
      VAR(2)=ZST(1)
      ELSE
      TYPE *,'INITIAL GUESS OF DIELECTRIC CONSTANT AT F(1)'
      ACCEPT *,VAR
```

```
      QLAST = QUER('START AT LAST VALUE?',20)
      IF(QLAST)THEN
      PRINT *,'ITERATION STARTS FROM PREVIOUS FREQ  SOLUTION'
      ELSE
      PRINT*,'ITERATION STARTS FROM INITIAL GUESS AT EACH FREQ.'
      END IF
      PRINT 18
      VARIG(1) = VAR(1)
      VARIG(2) = VAR(2)
      ENDIF
C
C     SET ACCURACY
C
C     OK = 1.E-4
C     OK = (SQRT(VAR(1)*VAR(1)+VAR(2)*VAR(2)))*1.E-4
      EI = CMPLX(0.,1.)
C
C     START LOOP
C
      DO 1 J = 1,NPTS
      FF = F(J)
      IF(QFIL)THEN
      L=1
17    DDF = ABS(FF-FST(L))
      IF(DDF.LT.DF)THEN
      VAR(1)=YST(L)
      VAR(2)=ZST(L)
      GO TO 19
      ELSE
      IF(L.GE.NPT)THEN
      TYPE *,'NO FILE STARTING VALUE FOUND -- STARTING FROM LAST VALUE'
      GO TO 19
      END IF
      L=L+1
      END IF
      GO TO 17
      END IF
19    IF (QTANDR)GO TO 20
      EIPH = EI*Z(J)*1.7453293E-2
      AMAGN = 10.**(-Y(J)/20.)
      EVAL = AMAGN*CEXP(EIPH)
      VAL(1) = REAL(EVAL)
      VAL(2) = AIMAG(EVAL)
      GO TO 21
20    VAL(1) = 10.**(-Y(J)/20.)
      VAL(2) = 10.**(-Z(J)/20.)
21    CALL SIMUL(2,VAR,OK,QOUT)
40    FORMAT(I5,5G14.5)
      PRINT 40,J,F(J),Y(J),Z(J),VAR
      TYPE 40,J,F(J),Y(J),Z(J),VAR
      Y(J) = VAR(1)
      Z(J) = VAR(2)
      IF(.NOT.QLAST.AND..NOT.QFIL)THEN
              VAR(1)=VARIG(1)
              VAR(2)=VARIG(2)
      END IF
1     CONTINUE
```

```
C
C       OUTPUT
C
        DF = F(2)-F(1)
        QPL = QUER('PLOTS? (TEK 4105)',17)
        IF(QPL)THEN
        CALL PLOTS
        CALL PLTSTF(NPTS,F,Y,1,1,'REAL EP',' FREQ.  ','REAL EP')
        CALL WNTCPY
        CALL PLTSTF(NPTS,F,Z,1,2,' IM EP  ',' FREQ.  ',' IM EP  ')
        CALL WNTCPY
        END IF
        TYPE *,'DIELECTRIC CONSTANT OUTPUT FILE'
C       TYPE *,'REAL PERMITTIVITY FILE OUT'
C       CALL XYOUT(NPTS,QNAME,A,B,C,F,Y,0)
C       PRINT *,'REAL PERMITTIVITY FILE NAME',QNAME
C       TYPE *,'IMAGINARY PERMITTIVITY FILE OUT'
C       CALL XYOUT(NPTS,QNAME,A,B,C,F,Z,0)
C       PRINT *,'IMAGINARY PERMITTIVITY FILE NAME',QNAME
        CALL FYOUTN(QNAME,2,NPTS,F(1),F(NPTS),DF,F,Y,Z)
        TYPE 23,QNAME
        PRINT 23,QNAME
23      FORMAT(/,' OUTPUT FILE NAME:  ',A25)
        STOP
        END
```

CURVE FITTING AT EACH FREQUENCY: REFPOSTTEK FOR

Program REFPOSTTEK Fits, real, imaginary or magnitude S-parameter post data at each frequency in a waveguide band. The initial value for the fit is specified as a complex constant employed at all frequencies or employed at only the initial frequency with subsequent starting values taken as the solution at the previous frequency fit. Alternately, the initial guess can be a permittivity file generated by another program. Program REFPOSTTEK is less sensitive to be initial guess than EPOST and will average over multiple data sets.

```
      PROGRAM REFPostTek

C*****************************************************

C       Least Squares fit of (Re, Im, Abs val) S11, S21
C       data on post sample in waveguide.  Need
C       at least three data inputs for operation.
C       For instance, the input could be
C       Re and Im S11 and S21 data for a post
C       sample run on the HP8510 run several times
C       to get an average response or data from
C       several samples of the same material at
C       different diameters.
C
C       Input Files: A = Waveguide width
C                    B = Sample Diameter
C                    C = 1 Re(refl)
C                        2 Im(refl)
C                        3 Abs(refl)
C                        4 Re(trans)
C                        5 Im(trans)
C                        6 Abs(trans)

C*****************************************************


C       BRING IN DIMENSION, COMMON, ETC.


        IMPLICIT BYTE(G)
        DIMENSION XIN(201),YIN(201),REEP(201),AIMEP(201),CHISAV(201)
        DIMENSION CHI(4),XP(2),YP(2)
        DIMENSION YYDAT(60,201)
        COMMON /FRQ/F(201)/HOW/WIDTH(60),DIAM(60),HOW(60)
        1/PAR/PAR/VAR/VAR(2),DAR(2)
        2/QNAME/QNAME(60),QNAMFL
        3/LOCAL/YDAT(60),YCAL(60)
        CHARACTER*25 QOUT,QPLTDS,QNAM,QNAME
        DATA QNAME/60*'                         '/


        TYPE *,'PROGRAM RefPostTek'
        PRINT *,'PROGRAM RefPostTek'
C       INCLUDE 'PLTDES.FOR'
        QPLT=QUER('PLOTS? (TEK4105)',17)
        IF(QPLT)CALL PLOTS

C
C
C       get input file names
C
C
C
5       CALL NAMES(NFLS,A1,A2,A3)
        QPUNT = .FALSE.
```

```
C       Bring in files

        DO 1 J = 1,NFLS
2       QNAM=QNAME(J)
        CALL XYNIN(NPTS,QNAM,WIDTH(J),DIAM(J),HOW(J),XIN,YIN,0)
        TYPE *,J,WIDTH(J),DIAM(J),HOW(J)


        IF(J.EQ.1) GO TO 4

        IF(NPTS.EQ.NPT)GO TO 4
        TYPE *,'FILES HAVE DIFFERING NUMBER OF POINTS!?'
        GO TO 5

4       NPT = NPTS
        DO 3 N=1,NPTS
        IF(J.EQ.1)F(N) = XIN(N)
        DUM = YIN(N)
3       YYDAT(J,N)=DUM

C       Plot input files

        IF(QPLT)THEN

                IF(J.EQ.1)THEN
                        XP(1) = XIN(1)
                        XP(2) = XIN(NPTS)
                        YP(1) = -1
                        YP(2) = 1
                CALL PLTSTF(2,XP,YP,-1,3,'S-Parameter',' Frequency ',QNAM)
                END IF

                CALL PLTSTF(NPTS,XIN,YIN,0,J,' ',' ',QNAM)
        END IF

1       CONTINUE

        IF(QPLT)CALL WNTCPY


C       Initial guess on Re AND Im Pts.


        TYPE *,'INITIAL GUESS ON Re(ep) and Im(ep)'
        ACCEPT *,VAR(1),VAR(2)
        TYPE *,VAR
        IF(VAR(1).EQ.0..AND.VAR(2).EQ.0.)THEN
62              CALL FYINN(QNAM,N,NPT,F1,F2,DF,XIN,REEP,AIMEP,0)
                IF(NPT.NE.NPTS)THEN
                        TYPE *,'File number of pts different from input files!'
                        TYPE *,'Try again'
                        GO TO 62
                END IF
                PRINT 61,QNAM
61              FORMAT(/,' Starting from ep file: ',A25,/)
        ELSE
                PRINT *,'Initial guess Re(ep)=',VAR(1),' Im(ep)=',VAR(2)
                DO K = 1,NPTS
```

```
                              REEP(K)=VAR(1)
                              AIMEP(K)=VAR(2)
                    END DO
          END IF

          TYPE *, 'TYPE increments'
          ACCEPT *, DAR(1), DAR(2)
          TYPE *, DAR
          NTERMS = 2
          QLAST = QUER('Start from last value?',22)
          IF(QLAST)PRINT *, 'Starting from previous solution.'
          PRINT *
C
          QPLP=.FALSE.
          IF(QPLT)QPLP=QUER('PLOT IN LOOP?',13)
C
C         START FREQUENCY / FIT LOOP
C

          NFREE = NFLS - NTERMS
C
C     SET UP FREQUENCY LOOP
C
          NPT = 0
          JFS = 1
          NFSTEP = 1
          JFFF = NPTS

C     FREQ OFFSET AND STEP

C         IF (ISENSE(6)) 18,18,19
19        TYPE *, 'STARTING FREQUENCY #='
          ACCEPT 22,JFS
          TYPE *, ' STARTING AT FREQUENCY',F(JFS)
22        FORMAT(I8)
          TYPE *, ' FREQUENCY STEP INCREMENT ='
          ACCEPT 27,NFSTEP
27        FORMAT(I8)
          TYPE *, 'INCREMENT =',NFSTEP

C     CLEAR CHI SQUARE ACCUMULATOR

18        DO 16 J = 1,4
16        CHI(J) = 0.

C     BEGIN FREQUENCY LOOP

          DO 40 JF = JFS,NPTS,NFSTEP
          NPT = NPT + 1
          IF(.NOT.QLAST)THEN
                    VAR(1) = REEP(JF)
                    VAR(2) = AIMEP(JF)
          END IF

C         DOWN LOAD DATA AT FRQ JF

          DO 11 K = 1,NFLS
          DUM = YYDAT(K,JF)
11        YDAT(K)=DUM
C
```

```
C       SET UP FIT LOOP
C
        ALAMDA = .001
        MODE = 0.
        FRQ = F(JF)
C
C       PLOT DATA IN LOOP
C
        IF(.NOT.QPLP)GO TO 20
        IF(((NPT-1)/5)*5+1.EQ.NPT)THEN
                NPC = 7
        CALL PLTSTF(NFLS,HOW,YDAT,1,-NPC,'        ',' How code ','   V   ')

        ELSE
                NPC = NPT - (NPT/5)*5 + 6
                CALL PLTSTF(NFLS,HOW,YDAT,0,-NPC,' ',' ',' ')
        END IF
C
C       TOP OF FIT LOOP
C
20      CONTINUE

        CALL RPOSTCUR(FRQ,ALAMDA,CHISQR,NFLS,NTERMS,0)

C   SEE IF CHI SQUARE IS CHANGING

        CHH = 0.
        DO 31 K = 2,4
        CHH = CHH + CHI(K)
        KKK = K - 1
31      CHI(KKK) = CHI(K)
        CHI(4) = CHISQR
        CHH = CHH + CHISQR
        IF (CHISQR.LT.1.E-30) GO TO 30
        CHH = ABS(CHH/(4.*CHISQR)-1.)
30      CONTINUE

32      IF (ISENSE(2).EQ.0) GO TO 33
                TYPE *,'CHISQR',CHISQR,'LAMBDA',ALAMDA,'Re(ep) and Im(ep)',VAR
                IF (ISENSE(4).EQ.0) GO TO 33
                        TYPE 34, (YDAT(K),YCAL(K),K = 1,NFLS)
34              FORMAT (4(2G13.4,4X))
33      CONTINUE

C   TEST CHI SQUARE

        IF ((CHH.LE.1.E-5).OR.(CHI(4).LE.1.E-6).OR.(ALAMDA.EQ.1.2E6))
        1 GO TO 15
        GO TO 20

C   END FIT LOOP

15      CONTINUE

C       PLOT AFTER FIT

        IF(.NOT.QPLP)GO TO 35
        NPC = NPC - 6
        CALL PLTSTF(NFLS,HOW,YCAL,0,NPC,'        ','        ','        ')
```

```
C       PRINT AFTER FIT

35      TYPE *, JF, CHISQR, F(JF), VAR

C       SAVE Re and Im ep and CHISQUARE

        REEP(NPT) = VAR(1)
        AIMEP(NPT) = VAR(2)
        CHISAV(NPT) = CHISQR

C       END FREQUENCY LOOP

40      CONTINUE


C       OUTPUT


50      CONTINUE
        CALL TYDAT
        CALL PRDAT
        PRINT *, 'PROGRAM REFPostTek OUTPUT'
        PRINT 51, QNAMFL
51      FORMAT(' INPUT NAME FILE =', X, A25)
        PRINT *
        PRINT 52, (QNAME(K), K=1, NFLS)
52      FORMAT(5(X, A25))
        PRINT 53
53      FORMAT(5X, '#', 10X, 'CHI-SQUARE', 4X, 'Freq. ', 10X, 'Re(ep)', 9X, 'Im(ep)')
        DO 54 J = 1, NPTS
54      PRINT *, J, CHISAV(J), XIN(J), REEP(J), AIMEP(J)

        IF(QPLT)THEN
                CALL PLTSTF(NPT, XIN, REEP, 1, 3, 'Re(ep)', ' FREQ.  ', ' Re(ep)  ')
                CALL WNTCPY
                CALL PLTSTF(NPT, XIN, AIMEP, 1, 1, 'Im(ep)', ' FREQ.  ', ' Im(ep)  ')
                CALL WNTCPY
        END IF

        PRINT *
        PRINT *, 'Permittivity output file'
        F1 = F(JFS)
        DF = F(JFS+NFSTEP)-F1
        CALL FYOUTN(QOUT, 2, NPT, F1, FRQ, DF, F, REEP, AIMEP)
        TYPE 55, QOUT
        PRINT 55, QOUT
55      FORMAT(' OUTPUT FILE NAME =', 2X, A25, X)

        CALL CODE(2)

        STOP
        END


        FUNCTION RPOSTFCT(XI, I)

        COMPLEX EP
        LOGICAL Q
```

```
      COMMON /VAR/VAR(2),DAR(2)
      1/HOW/WIDTH(60),DIAM(60),HOW(60)

      EP = CMPLX(VAR(1),VAR(2))
      W = WIDTH(I)*2.54E-3
      D = DIAM(I)*2.54E-3
      Q = .FALSE.
      IH = HOW(I)
      GO TO (1,2,3,4,5,6)IH

4     Q = .TRUE.
1     RPOSTFCT = REPOST(EP,W,D,XI,Q)
      GO TO 10

5     Q = .TRUE.
2     RPOSTFCT = AIPOST(EP,W,D,XI,Q)
      GO TO 10

6     Q = .TRUE.
3     RPOSTFCT = ABPOST(EP,W,D,XI,Q)

10    RETURN
      END



      SUBROUTINE RPOSTCUR(FRQ,ALAMDA,CHISQR,NFLS,NTERMS,MODE)

      REAL*16 ARRAY,WEIGHT,DET,ASQ,ALPHA,BETA
      COMMON /VAR/VAR(2),DAR(2)
     2/LOOP/ARRAY(2,2)
      3/DER/DERIV(2)
      5/LOCAL/YDAT(60),YCAL(60)
     DIMENSION BETA(2), ALPHA(2,2),YI(60)
     DIMENSION A(2),B(2), C(2)



C     INCLUDE 'EMUL.FOR'



C     IF (ISENSE(13).EQ.1) TYPE 140
C 140 FORMAT(' C')

      CALL ERRSNS
D     TYPE *, 'CUR IN', ALAMDA,CHISQR,NFLS,NTERMS,MODE

   11 NFREE = NFLS - NTERMS

     IF (NFREE) 13,13,20
   13 CHISQR = 0.
     GO TO 110
20    CONTINUE

      ITST = 0
      ICURCY = 0
```

```
61       DO 62 I = 1,NFLS
         YCAL(I) = RPOSTFCT(FRQ,I)
62       YI(I) = YDAT(I) - YCAL(I)

63       CHISQ1 = FCHISR(NFLS,NFREE,MODE)

   31 DO 34 J=1,NTERMS
      BETA(J) = 0.
      DO 34 K=1,J
   34 ALPHA(J,K) = 0.


   41 DO 50 I= 1,NFLS

         CALL FDER(FRQ,I,NTERMS)

                 DO 46 J=1,NTERMS
                         BETA(J) = BETA(J) + YI(I)*DERIV(J)
D                        TYPE *,J,BETA(J),YI(I),DERIV(J)

                 DO 46 K=1,J
                         ALPHA(J,K) = ALPHA(J,K) + DERIV(J)*DERIV(K)

46               CONTINUE

50       CONTINUE

51       DO 53 J=1,NTERMS

         DO 53 K=1,J

53                ALPHA(K,J) = ALPHA(J,K)


71       DO 74 J=1,NTERMS

                 DO 73 K=1,J
                         ASQ = SQRT(ALPHA(J,J)*ALPHA(K,K))
                         ARRAY(J,K) = ALPHA(J,K)/ASQ
73                       ARRAY(K,J) = ARRAY(J,K)

74               ARRAY(J,J)=1.+ALAMDA

   80 CALL MATIRR(NFLS,NTERMS,DET)

81       DO 86 J=1,NTERMS

                 A(J) = VAR(J)
                 B(J) = A(J)

         DO 86 K=1,NTERMS

86               B(J) = B(J)+BETA(K)*ARRAY(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))


47       DO 120 J = 1, NTERMS
                 C(J) = A(J)
D                TYPE *,B(J)
```

```
                A(J) = B(J)
                VAR(J) = A(J)
120       CONTINUE

91        DO 92 I=1,NFLS

              YCAL(I) = RPOSTFCT(FRQ,I)

92        CONTINUE

93        CHISQR = FCHISR(NFLS,NFREE,MODE)

          CALL ERRTST(10,ITST)

        IF (CHISQ1 - CHISQR) 95,101,101

95        ALAMDA = 10.*ALAMDA

          DO 130 J = 1, NTERMS
                A(J) = C(J)
                VAR(J) = A(J)
130       CONTINUE

          ICURCY = ICURCY + 1
          IF (ICURCY.GE.80) GO TO 102
          IF (ITST.EQ.1) GO TO 102
          IF (ALAMDA.GE.1E19)GO TO 102
          GO TO 71

102        CHISQR = CHISQ1

        ALAMDA = 1.2E7

  101 CONTINUE

        ALAMDA = ALAMDA/10.

  110 RETURN
      END


          SUBROUTINE FDER(FRQ,I,NTERMS)

C*****************************************************************

C   SUBROUTINE FDERIV (NON ANALYTICAL)

C   PURPOSE
C     EVALUATE DERIVATIVES OF FUNCTION FOR LEAST-SQUARES SEARCH
C       FOR ARBITRARY FUNCITON GIVEN BY FUNCTN

C   USAGE
C     CALL FDERIV (X,I,A,DAR,NTERMS,DERIV)

C   DESCRIPTION OF PARAMETERS
C     X       - ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
```

```
C      I       - INDEX OF DATA POINTS
C      A       - ARRAY OF PARAMETERS
C      DAA  - ARRAY OF PARAMETER INCREMENTS
C      NTERMS - NUMBER OF PARAMETERS
C      DERIV  - DERIVATIVES OF FUNCTION

C    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      FUNCTN (X, I, A)
C         EVALUATES THE FITTING FUNCTION FOR THE I TH TERM


C        MODIFIED FOR NET ANALYSIS -- SEE DERIVE & YFTE'S
C

C**************************************************U**U*U*****>**U



        COMMON /VAR/VAR(2), DAR(2)
        1/DER/DERIV(2)



        IF(ISENSE(13).EQ.1) TYPE 1
1       FORMAT('D')
C
C
   11 DO 18 J = 1, NTERMS
        AJ = VAR(J)
         DELTA = DAR(J)
        VAR(J) = AJ + DELTA
        YFT = RPOSTFCT(FRQ, I)
C
C
        VAR(J) = AJ - DELTA
        DERIV(J) = (YFT - RPOSTFCT(FRQ, I)) / (2.*DELTA)
C
C
        IF (DERIV(J).EQ.0.) DERIV(J) = .000001
18      VAR(J) = AJ
C
C
C
C
22      CONTINUE
        RETURN
        END
C   FUNCTION FCHISR
C
C    PURPOSE
C      EVALUATE REDUCED CHI SQUARE FOR FIT TO DATA
C         FCHISR = SUM((Y-YCAL)**2)/SIGMA**2)/NFREE
C
C    USAGE
C      RESULT = FCHISR (Y, SIGMAY, NFLS, NFREE, MODE, YCAL)
C
      FUNCTION FCHISR(NFLS, NFREE, MODE)
      DOUBLE PRECISION CHISQ
      COMMON /LOCAL/YDAT(60), YCAL(60)
      IF (ISENSE(13).EQ.1)TYPE 1
```

```
    1 FORMAT(' X')
   11 CHISQ = 0.
   12 IF (NFREE) 13,13,20
   13 FCHISR = 0.
      GO TO 40
C
C   ACCUMULATE CHI SQUARE
C
   20 CONTINUE
      DO 30 I = 1,NFLS
C    21 IF (MODE) 22,27,22
C    22 IF (ABS(YDAT(I)).LE.1.E-6) GO TO 27
C         IF (YDAT(I)) 25,27,23
C    23 WEIGHT = 1./YDAT(I)
C       GO TO 30
C    25 WEIGHT = 1./(-YDAT(I))
C       GO TO 30
C    27 WEIGHT = 1.
C     GO TO 30
C   29 WEIGHT = 1./SIGMAYDAT(I)**2
D        TYPE *,'CHI',CHISQ,WEIGHT(I),YDAT(I),YCAL(I)
 30      CHISQ = CHISQ + (YDAT(I)-YCAL(I))**2
C
C   DIVIDE BY NUMBER OF DEGREES OF FREEDOM
C
   31 FREE = NFREE
   32 FCHISR = CHISQ/(FREE)
   40 CONTINUE
D        TYPE2,NFREE,FCHISR
D2       FORMAT(I8,G13.5)
         RETURN
      END


         SUBROUTINE MATIRR(NPTS,NTERMS,DET)
      COMMON/LOOP/ARRAY(2,2)
         REAL*16 ARRAY,AMAX,SAV,DET
      DIMENSION IK(2),JK(2)
      NORDER = NTERMS
      IF (ISENSE(13).EQ.1) TYPE 1
    1 FORMAT(' M')
   10 DET = 1.
   11 DO 100 K= 1,NORDER
      AMAX = 0.
   21 DO 30 I = K, NORDER
      DO 30 J=K,NORDER
   23 IF (ABS(AMAX)-ABS(ARRAY(I,J))) 24,24,30
   24 AMAX = ARRAY(I,J)
      IK(K)= I
      JK(K)=J
   30 CONTINUE
   31 IF (AMAX ) 41,32,41
   32 DET = 0.
      GO TO 140
   41 I = IK(K)
      IF (I-K) 21,51,43
   43 DO 50 J=1,NORDER
      SAV = ARRAY (K,J)
      ARRAY (K,J) = ARRAY(I,J)
```

```
   50 ARRAY(I,J) = -SAV
   51 J = JK(K)
      IF (J-K) 21,61,53
   53 DO 60 I=1,NORDER
      SAV = ARRAY(I,K)
      ARRAY(I,K) = ARRAY(I,J)
   60 ARRAY(I,J) = -SAV
   61 DO 70 I=1,NORDER
      IF (I-K) 63,70,63
   63 ARRAY(I,K) = -ARRAY(I,K)/AMAX
   70 CONTINUE
   71 DO 80 I=1,NORDER
      DO 80 J= 1,NORDER
      IF (I-K) 74,80,74
   74 IF (J-K) 75,80,75
   75 ARRAY(I,J) = ARRAY(I,J) + ARRAY(I,K)*ARRAY(K,J)
   80 CONTINUE
   81 DO 90 J=1,NORDER
      IF (J-K) 83,90,83
   83 ARRAY(K,J) = ARRAY(K,J)/AMAX
   90 CONTINUE
      ARRAY(K,K) = 1./AMAX
D        TYPE *,' DET',DET,'AMAX',AMAX
  100 DET = DET*AMAX
  101 DO 130 L=1,NORDER
      K = NORDER -L + 1
      J = IK(K)
      IF (J-K) 111,111,105
  105 DO 110 I=1,NORDER
      SAV = ARRAY(I,K)
      ARRAY(I,K) = -ARRAY(I,J)
  110 ARRAY(I,J) = SAV
  111 I = JK(K)
      IF (I-K) 130,130,113
  113 DO 120 J = 1,NORDER
      SAV = ARRAY(K,J)
      ARRAY(K,J) = -ARRAY(I,J)
  120 ARRAY(I,J) = SAV
  130 CONTINUE
  140 RETURN
      END
```

FITTING MULTIPLE DATA SETS TO A FUNCTION OF FREQUENCY: FTPOST.FOR

The following subroutines are linked to a general fitting program to produce POSTTEK.

```
        SUBROUTINE FTOPT
C
C
C       FTDDOP.FOR
C
C
C
        IMPLICIT BYTE(Q)
        COMMON/OPTION/IAFCT(5),IBFCT(5),NRT,NCV(5),QIND,QOPT
        1/OPT2/QEPC,QQQ
        1/NUM/NPTS,NFLS,NTERMS,NPAR
        3/DBV/QDBTOV
        DIMENSION IFCT(25)
        CHARACTER*50 FCTLAB
        DATA IFCT/3,3,0,3,0,3,0,3,3,5,0,5,0,1,3,0,4,3,3,7,0,5,0,0,0/
C
C
C
        QOPT = .TRUE.
        QEPC = .FALSE.
        QDBTOV = .FALSE.
        QDBTOV = QUER('CONVERT DB DATA TO V?',27)
C
C
C
C
C       CALL FTANG(NFLS,NRT)
C
C
C
C
C
C
        DO 5 J = 1,23
        CALL FTFLAB(FCTLAB,J)
5       TYPE 300,FCTLAB
C
C
C
        NTOT = 0
        IBFCT(1) = 1

        DO 1 J = 1,2
        IF (J.EQ.1)THEN
                TYPE *,'REAL PERMITTIVITY'
                PRINT *,'REAL PERMITTIVITY'
        ELSE
                TYPE *,'IMAGINARY PERMITTIVITY'
                PRINT *,'IMAGINARY PERMITTIVITY'
        END IF

        TYPE *,'FUNCTION NUMBER'
        ACCEPT 2, IAFCT(J)
        IA = IAFCT(J)
2       FORMAT(I8)
        TYPE *,'FUNCTION #',IAFCT(J)
        CALL FTFLAB(FCTLAB,IAFCT(J))
        TYPE 300,FCTLAB
        PRINT *,'FUNCTION #',IAFCT(J)
        PRINT 300,FCTLAB
300     FORMAT(X,A60)
```

```
      NCV(J)=1
      IF (IA.NE.6.AND.IA.NE.12.AND.IA.NE.15.AND.IA.NE.22) GO TO 3
      TYPE *,'NUMBER OF CURVES'
      PRINT *,'NUMBER OF CURVES'
      ACCEPT 2,NCV(J)
      TYPE *,NCV(J)
      PRINT *,NCV(J)
      QEPC = QUER('ADD A COMPLEX CONST EP?',23)
C     QIND(IA) = QUER('INDEPENDENT AMPLITUDES?',23)

3     IF(IA.EQ.10.OR.IA.EQ.12)TYPE*,'AMPL,4PIM,K,LAM,GAM'
      IF(IA.EQ.22)TYPE *,'HEIGHT, ROLL OFF FRQ, RELAX FRQ, ALPHA, BETA'
      IF(IA.EQ.2.OR.IA.EQ.4.OR.IA.EQ.6.OR.IA.EQ.15)TYPE *,
     1'HEIGHT, CTR FRQ, WIDTH'

      NVARA = NCV(J)*IFCT(IA)
      IF(QEPC.AND.(IA.EQ.6.OR.IA.EQ.12.OR.IA.EQ.15.OR.IA.EQ.22))
     1NVARA = NVARA + 2
      TYPE *,NVARA,'PARAMETERS'
      NTOT = NTOT + NVARA
      IF (J.EQ.4) GO TO 4
      J1 = J + 1
      IBFCT(J1) = IBFCT(J) + NVARA
4     IF(NVARA.NE.0) GO TO 1
      JM1 = J - 1
      IBFCT(J) = IBFCT(JM1)
      NCV(J) = NCV(JM1)
C     QIND(J) = QIND(JM1)
1     CONTINUE
      TYPE *,'TOTAL PARAMETERS =',NTOT
      NPAR = NTOT
D     TYPE *,'IAFCT,IBFCT,NRT,NCV,QIND',IAFCT,IBFCT,NRT,NCV,QIND
      RETURN
      END
      SUBROUTINE FTFLAB(QFCT,J)
      CHARACTER*50 QFCTN
      CHARACTER*(*) QFCT
      QFCTN='                                                  '
      QFCT=QFCTN
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,114,115
     1,116,117,118,119,120,121,122,123)J
101   QFCT = ' FUNCTN 1: A + B*FRQ + C*FRQ**2'
      GO TO 1000
102   QFCT = ' FUNCTN 2: REAL LORENTZIAN + 1'
      GO TO 1000
103   QFCT = ' FUNCTN 3: IMAG LORENTZIAN'
      GO TO 1000
104   QFCT = ' FUNCTN 4: REAL GAUSSIAN'
      GO TO 1000
105   QFCT = ' FUNCTN 5: IMAGINARY GAUSSIAN'
      GO TO 1000
106   QFCT = ' FUNCTN 6: SEVERAL REAL GAUSSIANS'
      GO TO 1000
107   QFCT = ' FUNCTN 7: SEVERAL IMAGINARY GAUSSIANS'
      GO TO 1000
108   QFCT = ' FUNCTN 8: CONDUCTIVE LOSS'
      GO TO 1000
109   QFCT = ' FUNCTN 9: SPOT DIELECTRIC'
      GO TO 1000
```

```
110     QFCT = ' FUNCTN 10: REAL KITTEL PERMEABILITY'
        GO TO 1000
111     QFCT = ' FUNCTN 11: IMAGINARY KITTEL PERMEABILITY'
        GO TO 1000
112     QFCT = ' FUNCTN 12: SEVERAL REAL KITTEL'
        GO TO 1000
113     QFCT = ' FUNCTN 13: SEVERAL IMAGINARY KITTEL'
        GO TO 1000
114     QFCT = ' FUNCTN 14: CONSTANT'
        GO TO 1000
115     QFCT = ' FUNCTN 15: SEVERAL REAL LORENTZIANS'
        GO TO 1000
116     QFCT = ' FUNCTN 16: SEVERAL IMAGINARY LORENTZIANS'
        GO TO 1000
117     QFCT = ' FUNCTN 17: A + B*(FRQ-D) + C*(FRQ-D)**2'
        GO TO 1000
118     QFCT = ' FUNCTN 18: A + B*FRQ**C'
        GO TO 1000
119     QFCT = ' FUNCTN 19: A + B*EXP(FRQ*C)'
        GO TO 1000
120     QFCT = ' FUNCTN 20: PAR(1) + RE DAMPED HAR OSC(3,4,5,6,7)'
        GO TO 1000
121     QFCT = ' FUNCTN 21: PAR(2) + IMAG DAM HAR OSC'
        GO TO 1000
122     QFCT = ' FUNCTN 22: SEV RE DAMP HAR OSC(H,fo,fr,a,b)'
        GO TO 1000
123     QFCT = ' FUNCTN 23: SEV IM DAMP HAR OSC'
1000    RETURN
        END


        SUBROUTINE FTOPFN(I,N,VALU)
        IMPLICIT BYTE (Q)
C       IMPLICIT DOUBLE PRECISION (U)
        IMPLICIT COMPLEX (E)
        COMPLEX KITMU,DAMHAR
        COMMON/INDEP/X(1001),Y(1001),YFIT(1001)
        1/PAR/PAR(51)
        2/OPTION/IAFCT(5),IBFCT(5),NRT,NCV(5),QIND,QDUM
        3/OPT2/QEPC,QQQ
        DIMENSION CON(12)
C
C
C
        NTM = IBFCT(N)
        IGO = IAFCT(N)
        NCRV = NCV(N)
C       QINDD = QIND(IGO)
CD      TYPE *,IGO
        NTM2 = NTM + 1
        NTM3 = NTM + 2
        NTM4 = NTM + 3
        NTM5 = NTM + 4
        NTM6 = NTM + 5
        NTM7 = NTM + 6
        XI = X(I)
        GO TO (1,2,3,4,5,6,6,7,8,9,10,6,6,50,6,6,11,12,13,14,15,6,6) IGO
C
C   CONST
C
```

```
50        VALU = PAR(NTM)
C         IF (QCYC) NTM = NTM + 1
          GO TO 100
C
C    POLYNOMIAL
C
11        XI = XI - PAR(NTM4)
1         VALU = PAR(NTM) + PAR(NTM2)*XI + PAR(NTM3)*XI*XI
C         IF (QCYC) NTM = NTM + 3
          GO TO 100
C
C    REAL LORENTZIAN
C
2         VALU = 1. + ALOR(XI,PAR(NTM),PAR(NTM2),PAR(NTM3), TRUE.)
          GO TO 100
C
C    NO ADVANCE NTM
C
C    IMAGINARY LORENTZIAN
C
3         VALU = ALOR(XI,PAR(NTM),PAR(NTM2),PAR(NTM3),.FALSE.)
C         IF (QCYC) NTM = NTM + 3
          GO TO 100
C
C    REAL GAUSSIAN
C
4         VALU = 1. + COMGAU(XI,PAR(NTM),PAR(NTM2),PAR(NTM3),.TRUE.)
          GO TO 100
C
C    IMAGINARY GAUSSIAN
C
5         VALU = COMGAU(XI,PAR(NTM),PAR(NTM2),PAR(NTM3),.FALSE.)
C         IF (QCYC) NTM = NTM + 3
          GO TO 100
C
C    KITTEL PERMEABILITY
C
9         VALU = 1 + PAR(NTM)*
          1REAL(KITMU(XI,PAR(NTM2),PAR(NTM3),PAR(NTM4),PAR(NTM5)))
          GO TO 100
C
C         IMAGINARY KITTEL PERMEABILITY
C
10        VALU = PAR(NTM)*
          1AIMAG(KITMU(XI,PAR(NTM2),PAR(NTM3),PAR(NTM4),PAR(NTM5)))
C         IF (QCYC) NTM = NTM + 4
          GO TO 100
C
C    DAMPED HARMONIC OSCILLATOR
C
C    REAL
C
14        VALU = PAR(NTM) + REAL(DAMHAR(XI,PAR(NTM3),PAR(NTM4),PAR(NTM5),
          1PAR(NTM6),PAR(NTM7)))
          GO TO 100
C
C    IMAGINARY
C
15        VALU = PAR(NTM2)+AIMAG(DAMHAR(XI,PAR(NTM3),PAR(NTM4),PAR(NTM5),
          1PAR(NTM6),PAR(NTM7)))
```

```
C       IF (QCYC) NTM = NTM + 6
        GO TO 100
C
C   MULTIPLE GAUSSIANS
C
6       CALL FITMUL(XI,VALU,IGO,NTM,NCRV)
        GO TO 100
C
C   CONDUCTIVE LOSS
C
7       VALU = PAR(NTM) + 1800.*PAR(NTM2)/XI + PAR(NTM3)/(XI*SQRT(XI))
C       IF (QCYC) NTM = NTM + 3
        GO TO 100
C
C   SPOT DIELECTRIC - RUFUS WRIGHT
C
8       VALU = 4.*PAR(NTM)/(3.1415927*PAR(NTM2)*PAR(NTM3))
        GO TO 100
C
C   POWER LAW
C
12      VALU = PAR(NTM) + PAR(NTM2)*(XI**PAR(NTM3))
        GO TO 100
C
C   EXPONENTIAL
C
13      VALU = PAR(NTM) + PAR(NTM2)*EXP(XI*PAR(NTM3))
C
C
100     CONTINUE
D       TYPE *,XI,VALU
        RETURN
        END



        FUNCTION ALOR(XI,A,B,C,QREIM)
        IMPLICIT BYTE (Q)
        IF (.NOT.QREIM) GO TO 3
C
C   REAL LORENTZIAN
C
2       XX = B - XI
        D = 1./C
        ALOR = A*XX*D/(1+XX*XX*D*D)
        GO TO 100
C
C   IMAGINARY LORENTZIAN
C
3       XX = B - XI
        D = 1/C
        ALOR = A/(1+XX*XX*D*D)
100     RETURN
        END
```

```
      FUNCTION COMGAU(XI,A,B,C,QRE)
      IMPLICIT BYTE(Q)
      IMPLICIT DOUBLE PRECISION (U)
CD    TYPE *,XI,A,B,C,QRE
      IF (QRE) GO TO 4
      GO TO 5
C
C  REAL GAUSSIAN
C
4     XX = XI - B
      SIGRT2 = 0.6008*C
      XXBYS = XX/SIGRT2
      IF(ABS(XXBYS).GT.3) GO TO 402
      XXBYS2 = XXBYS*XXBYS
      UXBYS2 = XXBYS2
      SA = -1.12838*A*XXBYS
      UDSB = DEXP(-UXBYS2)
      ITM = XXBYS2 + 4
      UDSC = 1.
      DJN = 1.
      ACLUXX = 1.
      DO 401 J=1,ITM
      DJN = DJN * J
      ACLUXX = ACLUXX*XXBYS2
      SD = 2*J + 1
401   UDSC = UDSC + ACLUXX/(DJN*SD)
      COMGAU = UDSB*UDSC
      COMGAU = SA*COMGAU
      COMGAU = COMGAU + 0.7979*A*SIGRT2/(XI + B)
      GO TO 100
402   COMGAU = -1.12838*A*B/((B + XI)*XXBYS)
      GO TO 100
C
C  IMAGINARY GAUSSIAN
C
5     XX = XI - B
      SIGRT2 = 0.6*C
      XXBYS = XX/SIGRT2
      XXBYS2 = XXBYS*XXBYS
      COMGAU = A*EXP(-XXBYS2)
      IF(QCYC) NTM = NTM + 3
      GO TO 100
C
C
C
100   CONTINUE
CD    TYPE *,COMGAU
      RETURN
      END



      FUNCTION DAMHAR(F,H,FO,FR,DIS,BETA)
      COMPLEX DAMHAR,EI
      EI = CMPLX(0.,1.)
      A = H*FR*FO
      DAMHAR = A/(FO*FO - F*F - (EI*FR*F)**(1.-DIS))**BETA
```

```
      RETURN
      END


      FUNCTION KITMU(F,A1,K,LAM,GAM)
      REAL K,LAM
      COMPLEX KITMU,FAC1,FAC2
      A5=A1/(4.*3.1415927)
      T1=2.*K/A5
      T2=-F*LAM/(A5*GAM**2)
      FAC1=CMPLX((T1+A1),T2)
      FAC2=CMPLX(T1,T2)
      KITMU=A1*FAC1/(FAC2*FAC1-(F/GAM)**2)
      RETURN
      END


      FUNCTION KITM(F,A1,AK,ALAM,GAM,QRI)
C
C     REAL AND IMAGINARY KITTEL FCTS
C     VIL KITMU
C
      IMPLICIT BYTE(Q)
      REAL KITM
      COMPLEX KITMU
      IF (QRI) KITM = REAL(KITMU(F,A1,AK,ALAM,GAM))
      IF (.NOT.QRI) KITM = AIMAG(KITMU(F,A1,AK,ALAM,GAM))
D     TYPE *,'K',KITM,QRI
      RETURN
      END


      SUBROUTINE FITMUL(XI,VALU,IGO,NTM,NCRV)
      IMPLICIT COMPLEX (E)
      IMPLICIT BYTE (Q)
      COMMON /PAR/PAR(51)
     1/OPT2/QEPC,QQQ
      COMPLEX DAMHAR
C
C  MULTIPLE GAUSSIANS, LORENTZIANS, KITTEL FCTS, DHO'S
C
      QLOG = .FALSE.
      QKIT = .FALSE.
      QGAU = .FALSE.
      QDHO = .FALSE.
      IF (IGO.EQ.6.OR.IGO.EQ.7) GO TO 6
      IF (IGO.EQ.10.OR.IGO.EQ.11) GO TO 40
      IF (IGO.EQ.15.OR.IGO.EQ.16) GO TO 11
      IF (IGO.EQ.22.OR.IGO.EQ.23) GO TO 41
      GO TO 100
```

```
41      QDHO = .TRUE.
        KPAR = 5
        GO TO 401
40      QKIT = .TRUE.
        KPAR = 5
        GO TO 401
11      QLOR = .TRUE.
        GO TO 402
6       QGAU = .TRUE.
402     KPAR = 3

401     ASM = 0.
        BSM = 0.

        QTP = .TRUE.
        IF(IGO.EQ.7.OR.IGO.EQ.11.OR.IGO.EQ.16.OR.IGO.EQ.23) QTP = .FALSE.

36      DO 31 J = 1,NCRV
        JJ = J - 1
        NTEM = NTM + KPAR*JJ
        NTEM2 = NTEM + 1
        NTEM3 = NTEM2 + 1
        NTEM4 = NTEM3 + 1
        NTEM5 = NTEM4 + 1
        NTEM6 = NTEM5 + 1
403     AHT = PAR(NTEM)
        IF (QGAU)BSM = BSM + COMGAU(XI,AHT,PAR(NTEM2),PAR(NTEM3),QTP)
        IF (QLOR) BSM = BSM + ALOR(XI,AHT,PAR(NTEM2),PAR(NTEM3),QTP)
        IF (QKIT)
        1BSM=BSM+AHT*KITM(XI,PAR(NTEM2),PAR(NTEM3),PAR(NTEM4)
        2,PAR(NTEM5),QTP)
        IF (QDHO)THEN
                EDHO = DAMHAR(XI,AHT,PAR(NTEM2),PAR(NTEM3),PAR(NTEM4)
        1,PAR(NTEM5))
                        IF(QTP)THEN
                                DHO = REAL(EDHO)
                        ELSE
                                DHO = AIMAG(EDHO)
                        END IF
                BSM = BSM + DHO
        END IF
31      CONTINUE

32      IF (QTP) BSM = BSM + 1.

        IF (QEPC)THEN
                IF(QTP)THEN
                        BSM = BSM - 1. + PAR(NTM+KPAR*NCRV)
                ELSE
                        BSM = BSM + PAR(NTM+KPAR*NCRV+1)
                END IF
        END IF

        VALU = BSM

100     RETURN
        END
        SUBROUTINE FTANG(NFLS,NRT)
1       RETURN
        END
```

```
      SUBROUTINE FTLAB(QP)
      BYTE QP
      IF(QP)PRINT *, 'POST IN WAVEGUIDE ANALYSIS'
      TYPE *, 'POST IN WAVEGUIDE ANALYSIS'
      RETURN
      END
      FUNCTION FUN(I)
C
C
C
C
      IMPLICIT COMPLEX (E)
      IMPLICIT BYTE (Q)
      COMMON /INDEP/X(1001), Y(1001), YFIT(1001)
     1/CONTRL/AA(51), BB(51), CC(51)
     2/NUM/NPTS, NFLS, NTERMS, NPARV
     3/FILPT/NPT(51), NPTTSS(1001)
     4/PAR/PAR(51)
     5/OPTION/IAFCT(5), IBFCT(5), NRT, NCV(51), QIND(51), QOPT
     6/OFFSET/DD(51)
      XI = X(I)
      WIDTH = AA(NPTTSS(I))
      DIAM = BB(NPTTSS(I))
      ITYPE = CC(NPTTSS(I)) + .01
C
C
C
1     CALL FTOPFN(I, 1, A)
      CALL FTOPFN(I, 2, B)
      EP = CMPLX(A, B)
C     CALL FTOPFN(I, 3, A)
C     CALL FTOPFN(I, 4, B)
C     EMU = CMPLX(A, B)
C
C
C
C
C
      A = WIDTH*2.54E-3
     1D = DIAM*2.54E-3
C
C
C
      Q = .FALSE.
      GO TO (11, 12, 13, 14, 15, 16)ITYPE
14    Q = .TRUE.
11    FUN = REPOST(EP, A, D, XI, Q)
      GO TO 20
15    Q = .TRUE.
12    FUN = AIPOST(EP, A, D, XI, Q)
      GO TO 20
16    Q = .TRUE.
13    FUN = ABPOST(EP, A, D, XI, Q)
20    CONTINUE
      RETURN
```

FITTING EPOST/REFPOSTTEK PERMITTIVITY TO A FUNCTION: FTEM.FOR

The subroutine package FTEM is very similar to FTPOST listed above. In the case of FTEM the permittivity files are created by EPOST or REFPOSTTEK are fitted to a functional form. Since FTEM and FTPOST are similar, we list just the difference file below.

```
************
File SYS$USERDISKB:[RACHFORD]FTEM.FOR;47
   21             QDBTOV = QUER('CONVERT DB DATA TO VOLTAGE?',27)
   22   C
******
File SYS$USERDISKB:[RACHFORD]FTPOST.FOR;24
   21             QDBTOV = QUER('CONVERT DB DATA TO V?',27)
   22   C
************
************
File SYS$USERDISKB:[RACHFORD]FTEM.FOR;47
  488   300       FORMAT(' FUNCTIONAL FIT TO EP OR MU DATA')
  489             IF(QP)PRINT 300
  490             TYPE 300
  491             RETURN
******
File SYS$USERDISKB:[RACHFORD]FTPOST.FOR;24
  488             IF(QP)PRINT *, 'POST IN WAVEGUIDE ANALYSIS'
  489             TYPE *, 'POST IN WAVEGUIDE ANALYSIS'
  490             RETURN
************
************
File SYS$USERDISKB:[RACHFORD]FTEM.FOR;47
  514             IF (ITYPE.EQ.1) THEN
  515                     CALL FTOPFN(I,1,A)
  516                     FUN = A
  517             ELSE
  518                     CALL FTOPFN(I,2,B)
  519                     FUN = B
  520             END IF
  521             RETURN
  522             END
  523             SUBROUTINE FTSPOUT
******
File SYS$USERDISKB:[RACHFORD]FTPOST.FOR;24
  513   1         CALL FTOPFN(I,1,A)
  514             CALL FTOPFN(I,2,B)
  515             EP = CMPLX(A,B)
  516   C         CALL FTOPFN(I,3,A)
  517   C         CALL FTOPFN(I,4,B)
  518   C         EMU = CMPLX(A,B)
  519   C
  520   C
  521   C
  522   C
  523   C
  524             A = WIDTH*2.54E-3
  525             D = DIAM*2.54E-3
  526   C
  527   C
  528   C
  529             Q = .FALSE.
  530             GO TO (11,12,13,14,15,16)ITYPE
  531   14        Q = .TRUE.
  532   11        FUN = REPOST(EP,A,D,XI,Q)
  533             GO TO 20
  534   15        Q = .TRUE.
  535   12        FUN = AIPOST(EP,A,D,XI,Q)
  536             GO TO 20
  537   16        Q = .TRUE.
```

```
13      FUN = ABPOST(EP,A,D,XI,Q)
20      CONTINUE
        RETURN
        END
C       SUBROUTINE FTSPOUT
C       RETURN
C       END
        SUBROUTINE FTSPOUT
************

Number of difference sections found: 3
Number of difference records found: 36

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=LP:[RACHFORD]FTEM.DIF;-
    SYS$USERDISKB:[RACHFORD]FTEM.FOR;47-
    SYS$USERDISKB:[RACHFORD]FTPOST.FOR;24
```